Pre-Synopsis Report

On

Load Balancing in Cloud Computing

Under the supervision of

Prof. Dr. V K Jain
Dean –SET
MUST, Lakshmangarh

Submitted by

Mr. Manmohan Sharma
Enrollment No. - 120485

June, 2019

Department of CSE, School of Engineering and Technology

Mody University of Science and Technology

Lakshmangarh, Sikar, Rajasthan-332311

MODY UNIVERSITY
A Leading Women's University

Mody University of Science and Technology
School of Engineering & Technology
Lakshmangarh – 332311, Distt. Sikar
Phones: (01573) 225001 to 225012 (12 lines) Fax: (01573) 225042
http://www.modyuniversity.ac.in

DECLARATION

I, **Manmohan Sharma**, hereby declare that the work presented in this thesis entitled **"Load Balancing in cloud Computing",** has been carried out by me under the supervision of **Dr. V.K. Jain**, Professor & Dean, School of Engineering for the award of Degree of Doctor of Philosophy in **Engineering**, **Department of Computer Science and Engineering, School of Engineering and Technology, Mody University of Science and Technology, Lakshmangarh, (Rajasthan) India.**

To the best of my knowledge, the work is original and has not been submitted elsewhere for any other academic attainment.

I declare that I have faithfully acknowledged, given credit to and referred to the authors/research workers wherever their works have been cited in the text and the body of the thesis. I further certify that I have not willfully lifted up some other's work, results, text, para, figures, data etc. reported in books, journals, magazines, dissertations, reports, thesis etc., or available at websites and included them in this Ph. D. thesis and cited as my own work.

Date:

Place:

Manmohan Sharma
Enrolment No.: 120485

MODY UNIVERSITY
A Leading Women's University

Mody University of Science and Technology
School of Engineering & Technology
Lakshmangarh – 332311, Distt. Sikar
Phones: (01573) 225001 to 225012 (12 lines) Fax: (01573) 225042
http://www.modyuniversity.ac.in

CERTIFICATE

This is to certify that **Mr. Manmohan Sharma** has successfully completed the research work, entitled **"Load Balancing in cloud Computing"** and has submitted the thesis as a partial fulfillment for the award of the degree of Ph.D. in Engineering, Mody University of Science and Technology, Lakshmangarh. To the best of my knowledge, the work is original and has not been submitted elsewhere for any other academic attainment.

It is further certified that **Mr. Manmohan Sharma** has fulfilled all the requirements laid down by the Ph.D. ordinance, Mody University of Science and Technology, Lakshmangarh for the submission of the thesis.

13/06/2017

Supervisor
Dr. V.K. Jain
Professor & Dean,
School of Engineering and Technology,
Mody University of Science and Technology,
Lakshmangarh, Rajasthan, India

ACKNOWLEDGMENT

Firstly, I would like to express my sincere gratitude to my research guide Dr. VK Jain for his continuous support in my Ph.D. study and related research, for his patience, motivation and knowledge.

My sincere thanks to Dr. VK. Jain, Dean - SET, for his encouragement, motivation and support throughout my journey. Without his precious support it would not be possible to conduct this research work.

I would also like to thank Dr. A. Senthil, HOD- CSE Department, for his feedback, cooperation and suggestions. I am grateful to Dr. Anand Sharma, Dr. Niranjan Lal and Mr. Somil Jain for their insightful comments, feedback, encouragement and friendship.

I would also like to thank the Library and Laboratory staff of SET, (MUST) for extending their help at various occasions.

Last but not the least I would like to thank my family and friends for their love and support throughout this journey and my life in General.

Mr. Manmohan Sharma

ABSTRACT

Cloud computing is up and coming era of registering what's more, a creating processing worldview in the present-day industry, either might be government associations or people in general associations. In basic terms we will outline Cloud computing is assortment of assorted servers that take into consideration want of numerous customers in lightweight of their demands. Clouds have very capable knowledge centers to handle expansive variety of client's demands. Cloud is defined as a shared reservoir of dynamic resources and virtualization.

Load Balancing (LB) is needed to cater the needs of customer in an efficient manner. In LB the workload is distributed between numerous virtual machines (VM's) on a Server over the system, to accomplish ideal resource utilization, diminish data processing time, diminish in normal response time, and stay away from over-burden.

This research work focuses on providing a better load balancing solution to maximize the resource utilization. Resource availability is one of the major issues related to cloud adoption. Sharable availability of resources from cloud also put constraints in term of response time. These issues will be discussed and hybrid algorithm would be proposed to get better results in clouds.

The objective of this research work is to get better solution from load balancing and re-balancing approaches. Apart from this, analysis of other load balancing algorithms both static and dynamic will be done.LB with prioritization of users will also be done to provide better services to the end users.

Finally a load balancing approach is designed by combining Equally Spread Current execution (ESCE) and priority algorithm to get hybrid algorithm which performs better than priority algorithm. A result also shows it is better than other load balancing algorithm.

Table of Content

	Title	Page No.
	Acknowledgement	i
	Abstract	ii
	List of Figures	vi
	List of Tables	vii
Chapter 1.	**INTRODUCTION**	**1-13**
1.1	Cloud Computing Evolution	**1**
	1.1.1. Features of cloud Computing	**3**
	1.1.2. Cloud Computing Deployment Models	**5**
	1.1.3. Cloud Computing Service Models	**7**
1.2	Research issues in cloud Computing	**9**
1.3	Research Motivation	**11**
1.4	Organization of Thesis	**12**
	References	**13**
Chapter 2.	**LOAD BALANCING IN CLOUD**	**14-23**
2.1	Virtualization Technology	**14**
2.2	Virtualization in cloud	**16**
2.3	Load Balancing in Cloud	**17**
	2.3.1. Static Load Balancing Algorithms	**18**
	2.3.2 Dynamic Load Balancing Algorithms	**20**
2.4	Challenges of Load Balancing in cloud	**22**
	References	**23**
Chapter 3.	**REVIEW OF LITERATURE**	**24-32**
3.1	Design Issues related to good Load Balancing Algorithms	**24**
	3.1.1. Load estimation policy	**24**
	3.1.2. Load transfer policy	**24**
	3.1.3. Location policy	**25**
	3.1.4. State Information Exchange policy	**25**
	3.1.5. Priority assignment policy	**26**
3.2	Literature Analysis	**27**
3.3	Research Objectives	**30**

References 31

Chapter 4. LOAD RE-BALANCING APPROACH FOR DISTRIBUTED LOAD IN CLOUD ENVIRONMENT 33-39

4.1 Introduction 33

4.2 Load Balancing 33

4.3 Load Re-balancing Approach 35

4.4 Result and Discussion 37

4.5 Conclusion 38

References 39

Chapter 5. LOAD BALANCING IN CLOUD USING PRIORITIZATION BASED ON QUALITY OF SERVICES(QoS) DEMAND 40- 53

5.1 Introduction 40

5.2 Load Balancing 41

5.3 Prioritization Methodology 42

5.4 Proposed Algorithm 45

5.5 Results and Discussion 50

5.6 Conclusion 51

References 52

Chapter 6. DESIGN OF LOAD BALANCING APPROACH FOR BETTER RESOURCE UTILIZATION 54-64

6.1 Introduction 54

6.2 Existing Load Balancing Algorithm 55

6.2.1 Equally Spread Current Execution Algorithm 55

6.2.2 Priority Algorithm 55

6.3 Hybrid Algorithm 56

6.4 Simulation Environment 57

6.4.1 Cloud SIM 58

6.4.2 Eclipse 59

6.5 Results and Discussion 59

6.6 Conclusion 62

References 63

Chapter 7. CONCLUSION AND FUTURE WORK **65-66**
7.1 Conclusion
7.2 Future Scope

LIST OF FIGURES

Figure No.	Description	Page No.
1.1	Evolution of cloud computing	1
1.2	Cloud Computing main aspect and delivery model	4
1.3	Cloud Computing deployment Models	7
1.4	Cloud Service Models	9
2.1	Before virtualization	15
2.2	After Virtualization	15
2.3	Classification of Load Balancing Algorithm	18
4.1	Radio Energy Dissipation Model	34
4.2	Simulation Results	37
5.1	Flowchart of phase 1	47
5.2	Flowchart of phase 2	49
5.3	Graph showing improvement in response time or priority users	50
6.1	Load balancer with algorithms	57
6.2	Flowchart of hybrid algorithm	58
6.3	VM's in increasing order of MIPS	60
6.4	Performing sorting on cloudlets	60
6.5	All VM's are equally utilized	60
6.6	Output of Execution	61
6.7	Results comparison using line graph and bar graph	61

LIST OF TABLES

Table No.	Description	Page No.
4.1	Initial User with their loads	35
4.2	Initial Load distribution when VM's are idle.	35
4.3	Load distribution on VM's after rebalancing	36
5.1	Priority assignment based on CBQR	42
5.2	Priority assignment based on number of VMs	42
5.3	Priority assignment to Datacenter	43
5.4	Priority table of users	43
5.5	Data center in Asia Region	43
5.6	Data center in Africa Region	44
5.7	Data center in America Region	44
5.8	Response Time for user base served with same and different GL with some priority.	50
6.1	Result comparison with other algorithms	62

Chapter 1

INTRODUCTION

This chapter introduces Cloud computing, its evolution and various related technologies like distributed computing, grid computing, Cloud characteristics and the services offered by Cloud along with motivation of research and organization of thesis.

1.1 Cloud Computing Evolution

The term Cloud has been used historically as a metaphor for the Internet. Cloud computing has evolved through a number of phases which include Grid and Utility computing, Application Service Provision (ASP), and Software as a Service (SaaS) as shown in Figure1.1. The underlying concept of Cloud computing dates back to the 1960s, when John McCarthy opined that computation may someday be organized as a public utility. It is also said that Cloud computing concept of showing the entire relevant computer network was forwarded by J.C.R Licklider and John McCarthy. The idea of an "intergalactic computer network" was introduced in the sixties by J.C.R. Licklider, who

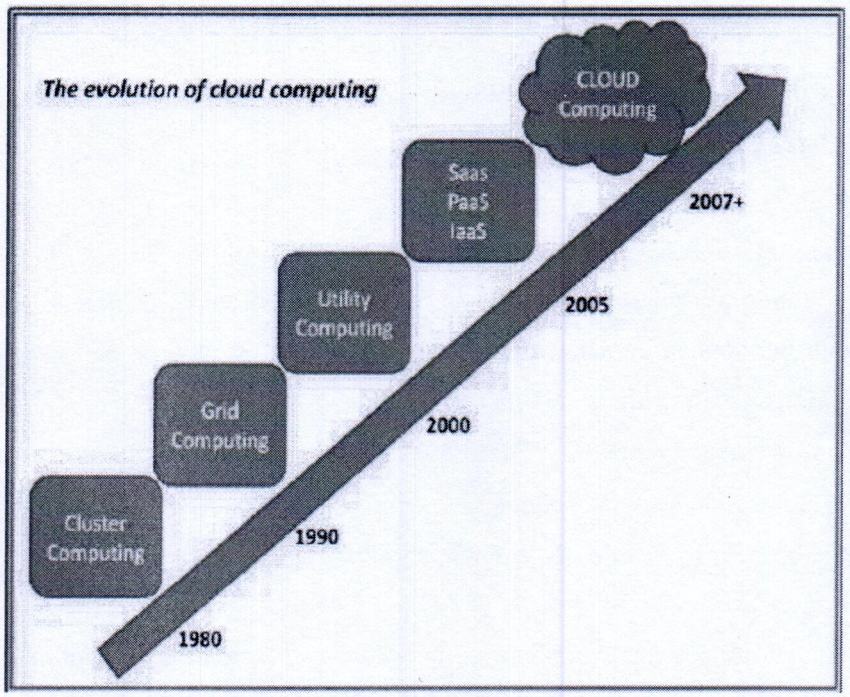


Figure 1.1: Evolution of cloud computing [3]

was responsible for enabling the development of ARPANET (Advanced Research Projects Agency Network) in 1969, his vision, was for everyone on the globe to be interconnected and accessing programs and data at any site, from anywhere. —Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services [1].
The data in a Cloud is stored on to centralized location called data centers having a large size of data storage. The data as well as processing is somewhere on servers. So, the clients have to trust the provider on the availability as well as data security. The SLA is the only legal agreement between the service provider and client. —The basic principle of Cloud computing is to distribute the computing tasks to many distributed computers, not local computer or remote servers [2].

In 1999 salesforce.com put the idea of Cloud computing in an application. They used a simple website to deliver its enterprise applications to users. Then in 2002 amazon launched their Cloud based web services. Cloud computing does not stop here it was then used by Microsoft to promote the Cloud computing services by Azure Services platform in October 27, 2008. Google in 2009 came into market by using Cloud computing concept by delivering Google Apps.

Cloud Computing offers service to end-users rather than a product, by sharing resources, software and other information under a usage based payment model. —Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services. Figure 1.2 shows the main aspects of Cloud computing. There are many proposed definitions of the Cloud computing due to its growing popularity defining its characteristics. Some of the definitions given by many well-known scientists and organizations are:

Rajkumar Buyya defines the Cloud computing in terms of its utility to end user as —A Cloud is a type of parallel and distributed system consisting of a collection of

interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers [4].

National Institute of Standards and Technology (NIST) defines Cloud computing as follows: —Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This Cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models [5].

1.1.1 Features of cloud Computing

The main features of the cloud computing are:
Broad network access: Cloud Services are available over the network and accessed through standard mechanisms that enhance the use by heterogeneous thin or thick client platforms (e.g. tablets, laptops, mobile phones and desktops).

Resource pooling: The cloud service provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to demand of the consumer.

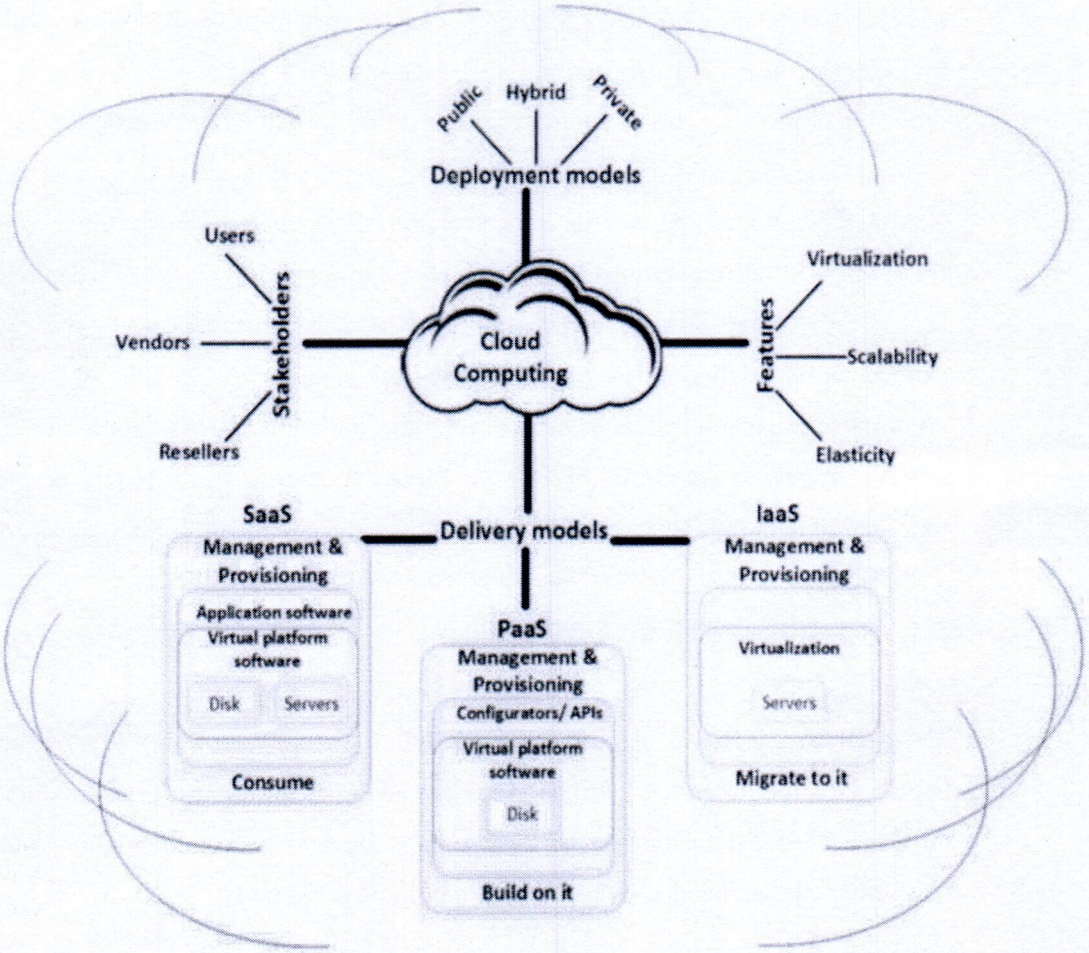


Figure1.2. Cloud Computing main aspect and delivery model [6].

Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward according to the demand. Consumers who is availing cloud services can increase and decrease their resources as per their needs.

Measured service: Cloud service provider can automatically control and optimize resource use by leveraging a metering capability for appropriate type of service (e.g., storage, processing, and bandwidth,). Resource usage can be monitored and reported, providing transparency for both the provider and consumer.

1.1.2. Cloud Computing Deployment Models

Different deployment models [7] of cloud computing are shown in figure 1.3 are as follows

❖ **Public cloud**

Public clouds are the most common way of deploying cloud computing. The cloud resources (like servers and storage) are owned and operated by a third-party cloud service provider and delivered over the Internet. In this the software, hardware and other supporting infrastructure are managed by cloud service provider. In a public cloud, hardware and storage space is shared among number of organizations or cloud "tenants." You access services and manage your account using a web browser. Public cloud deployments are frequently used to provide web-based email, online office applications, storage and testing and development environments.

Advantages of public clouds:

- Lower costs: Minimum cost is associated with this deployment model as there is no need to buy computing elements. You pay for the services you used.
- No maintenance: Maintenance is done by the service provider.
- Scalability: Highly scalable as per the business demands.
- High reliability: Highly reliable as too many servers are serving the customers.

❖ **Private Cloud**

A private cloud consists of computing resources used exclusively by one business or organization. The private cloud can be physically located at your organization's on-site datacenter or it can be hosted by a third-party service provider. But in a private cloud, the services and infrastructure are always maintained on a private network and the hardware and software are dedicated solely to your organization. In this way, a private cloud can make it easier for an organization to customize its resources to meet specific IT requirements. Private clouds are often used by government agencies, financial institutions, any other mid- to large-size organizations with business-critical operations seeking enhanced control over their environment.

Advantages of a private cloud:

- More flexibility—your organization can customize its cloud environment to meet specific business needs.
- Improved security—resources are not shared with others, so higher levels of control and security are possible.
- High scalability—private clouds still afford the scalability and efficiency of a public cloud.

❖ **Hybrid Cloud**

Often called "the best of both worlds," hybrid clouds combine on-premises infrastructure, or private clouds, with public clouds so organizations can reap the advantages of both. In a hybrid cloud, data and applications can move between private and public clouds for greater flexibility and more deployment options. For instance, you can use the public cloud for high-volume; lower-security needs such as web-based email and the private cloud (or other on-premises infrastructure) for sensitive, business-critical operations like financial reporting. In a hybrid cloud, "cloud bursting" is also an option. This is when an application or resource runs in the private cloud until there is a spike in demand (such as seasonal event like online shopping or tax filing), at which point the organization can "burst through" to the public cloud to tap into additional computing resources.

Advantages of hybrid clouds:

- Control—your organization can maintain a private infrastructure for sensitive assets.
- Flexibility—you can take advantage of additional resources in the public cloud when you need them.
- Cost-effectiveness—with the ability to scale to the public cloud, you pay for extra computing power only when needed.
- Ease—transitioning to the cloud does not have to be overwhelming because you can migrate gradually—phasing in workloads over time.

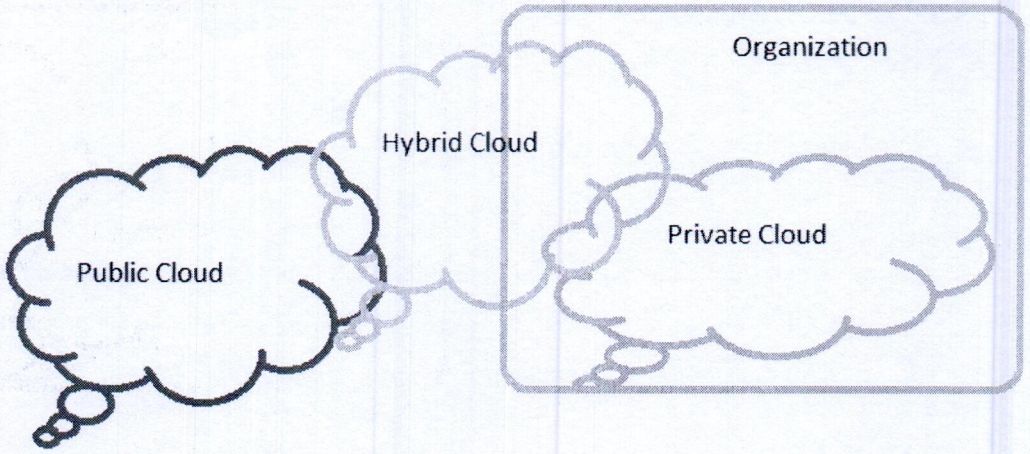


Figure 1.3: Cloud Computing deployment Models

1.1.3. Cloud Computing Service Models

Cloud generally has three basic types of service models as shown in figure 1.4 for offering different types of services to the cloud users [8]. These service models are as follows:

❖ **Software as a Service (SaaS)**

SaaS is a cloud computing offering that provides users with access to a vendor's cloud-based software. Users do not install applications on their local devices. Instead, the applications reside on a remote cloud network accessed through the web or an API. Through the application, users can store and analyze data and collaborate on projects.

Key features

- SaaS vendors provide users with software and applications via a subscription model.
- Users do not have to manage, install or upgrade software; SaaS providers manage this.
- Data is secure in the cloud; equipment failure does not result in loss of data.

- Use of resources can be scaled depending on service needs.
- Applications are accessible from almost any internet-connected device, from virtually anywhere in the world.

❖ **Platform as a Service (PaaS)**

PaaS is a cloud computing offering that provides users with a cloud environment in which they can develop manage and deliver applications. In addition to storage and other computing resources, users are able to use a suite of prebuilt tools to develop, customize and test their own applications.

Key features

- PaaS provides a platform with tools to test, develop and host applications in the same environment.
- Enables organizations to focus on development without having to worry about underlying infrastructure.
- Providers manage security, operating systems, server software and backups.
- Facilitates collaborative work even if teams work remotely.

❖ **Infrastructure as a Service (IaaS)**

PaaS is a cloud computing offering in which a vendor provides users access to computing resources such as servers, storage and networking. Organizations use their own platforms and applications within a service provider's infrastructure.

Key features

- Instead of purchasing hardware outright, users pay for IaaS on demand.
- Infrastructure is scalable depending on processing and storage needs.
- Saves enterprises the costs of buying and maintaining their own hardware.
- Because data is on the cloud, there can be no single point of failure.
- Enables the virtualization of administrative tasks, freeing up time for other work

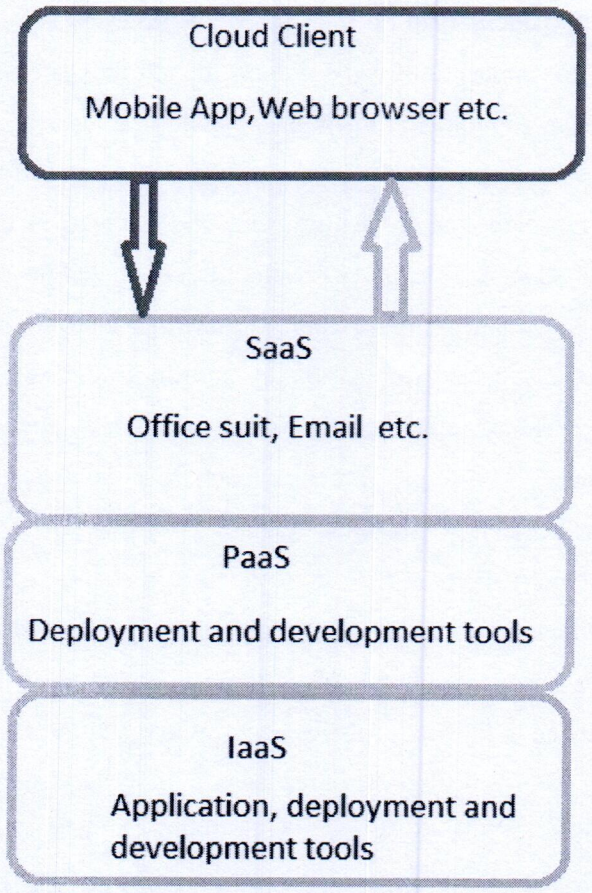


Figure1.4. Cloud Service Models

1.2. Research issues in cloud Computing

The evolution of cloud computing over the past few years is potentially one of the major advances in the history of computing. From the past few years , the cloud computing has made a tremendous impact on the Information Technology (IT) industry, where large companies such as Google, IBM, Amazon and Microsoft struggle to provide more powerful, reliable and cost-efficient cloud platforms, and business enterprises seek to find new paradigm in their business models. Apart from the advantages there are still some issues related to cloud that need to be discussed here:

A. Automated Service Provisioning :

The objective of a service provider in this case is to allocate and de-allocate resources from the Cloud to satisfy its service level objectives (SLOs), while

minimizing its operational cost. These approaches typically involve: (i) Constructing an application performance model that predicts the number of application instances required to handle demand at each particular level, in order to satisfy QoS requirements; (ii) Periodically predicting future demand and determining resource requirements using the performance model; and (iii) Automatically allocating resources using the predicted resource requirements. Application performance model can be constructed using various techniques, including Queuing theory, Control theory and Statistical Machine Learning. The proactive approach uses predicted demand to periodically allocate resources before they are needed. The reactive approach reacts to immediate demand fluctuations before periodic demand prediction is available.

B. Virtual Machine migration:

Virtual Machine migration provides major benefit in cloud computing through load balance across data centers. When users flood in cloud data centers, how to efficiently manage hardware resources and virtual machines (VMs) in a data center to both lower economical cost and ensure a high service quality becomes an inevitable work for cloud providers. VM migration [9] is a cornerstone technology for the majority of cloud management tasks. It frees a VM from the underlying hardware. This feature brings a plenty of benefits to cloud providers and users. VM migration schemes are classified from three perspectives: 1) manner; 2) distance; and 3) granularity. The studies on non-live migration are simply reviewed, and then those on live migration are comprehensively surveyed based on the three main challenges it faces: 1) memory data migration; 2) storage data migration; and 3) network connection continuity.

C. Data Security:

Clouds provide companies are still concerned about security when using cloud computing. Users are also worried about the vulnerability to attacks, when information and critical IT resources are outside the firewall. Where is the data more secure, on local hard drive or on high security servers in the cloud? However, in the cloud, the data will be distributed over the network through individual computers regardless of where the repository of data is ultimately stored.

Industrious hackers can invade virtually at any server, and there are the statistics show that one-third of breaches result from stolen or lost laptops and other devices and from employees' accidentally exposing data on the Internet, with nearly 16 percent due to insider stealing [10].

D. Energy Management :

Reducing the energy used in Cloud Computing is an important issue a sustainable society. There are many existing approaches for reducing energy use in data centers, but new approaches are needed in case of energy management of Cloud Computing. Cloud Computing involves decentralized data centers, so new flexible way for collecting energy consumption data becomes quite important. Many current approaches focus on reducing energy consumption by air handling equipment, however energy from IT resources also need to be optimized for a total energy management of Cloud Computing[11].

1.3. Research Motivation

Human reliance on cloud is clear from the way that today's most well-known social organizing, email, archive sharing and web based gaming destinations are facilitated on cloud. Google, Microsoft, IBM, Amazon, Yahoo and Apple among others are exceptionally dynamic in this field. As the companies are adopting cloud at a very fast pace it is one of the major concern to provide them better services. To cater enormously increasing user demands, service provider needs to think about balancing the load among the available resources.

There are many issues in Cloud computing such as Automated Service Provisioning, Virtual Machine Migration, Energy Management, Server Consolidation, Data Security, etc. as discussed in previous section that have not been fully addressed. The main focus of this thesis is load balancing in cloud computing. It is the way toward reassigning the aggregate loads to the individual nodes of the aggregate framework to make the best response time and furthermore great usage of the assets. It tackles situations where a portion of the nodes are over loaded or under loaded.

Various existing techniques manage the heterogeneous workloads but have some issues related to response time and others. Aim of this thesis to design and implement good load balancing algorithms to improve response time and other metrics related to cloud computing.

1.4 Organization of Thesis

The rest of the thesis is organized as follows:

Chapter 2 – This chapter describes in detail the literature survey done to study the concept of virtualization, load balancing techniques and cloud framework.

Chapter 3 – This chapter describes the problem Analysis of the thesis work. It gives the gap analysis and problem statement.

Chapter 4 – This chapter focus on the implementation details and experimental results – description of CloudSim, Eclipse and snapshots of the Cloud portal designed to study the technique.

Chapter 5 – This chapter describes the conclusion, contribution to the work done and future research work possible.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G.Lee, D. A. Patterson, A. Rabkin, I. Stoica and M. Zaharia, —Above the Clouds: A Berkeley View of Cloud Computing, University of California at Berkeley, Electrical Engineering and Computer Sciences, Technical Report No. UCB/EECS-2009-28, February 10, 2009.

[2] S. Zhang, S. Zhang, X. Chen and X. Huo, —The Comparison Between Cloud Computing and Grid Computing, International Conference on Computer Application and System Modeling, 2010.

[3] Ps Yoganandani, Rahul Johari et. Al, "Clearing The Clouds On Cloud Computing: Survey Paper" International Journal of Recent Development in Engineering and Technology. Vol-1,Issue-1,2014

[4] R. Buyya, C. Yeo, and S. Venugopal, —Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities, in Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-OB, IEEE CS Press. Los Alamitos, CA. USA), 2008.

[5] P. Mell, and T. Grance, —The NIST Definition of Cloud computing,‖ National Institute of Standards and Technology, 2009.

[6] Salman Iqbal, Miss Laiha Mat Kiah et.al, "Service delivery models of cloud computing: security issues and open challenges", SECURITY AND COMMUNICATION NETWORKS, Wiley Online Library (wileyonlinelibrary.com), pp 4726-4750, 2016.

[7] V. R. Srinivasa, N. K. R. Nageswara and E. K. Kumari, —Cloud Computing: An Overview,Journal of Theoretical and Applied Information Technology, 2009.

[8] https://www.ibm.com/cloud/learn/iaas-paas-saas

[9] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt and A. Warfield, —Live migration of virtual machines, In the Proceeding of NSDI, 2005.

[10] M.Rajendra Prasad, R. Lakshman Naik, V.Bapuji, "Cloud Computing : Research Issues and Implications" Vol.2, No.2, April 2013, pp. 134~140

[11] Fumiko Satoh , Hiroki Yanagisawa , Hitomi Takahashi ,Takayuki Kushida ," Total Energy Management System for Cloud Computing" IEEE International Conference on Cloud Engineering (IC2E), 13 June 2013,CA,USA

Chapter 2
LOAD BALANCING IN CLOUD

This chapter discusses the state of the art and research issues of Virtualization, Load balancing and rebalancing in cloud computing.

2.1 Virtualization

The term virtualization broadly describes the separation of a resource or request for a service from the underlying physical delivery of that service. With virtual memory, for example, computer software gains access to more memory than is physically installed, via the background swapping of data to disk storage. Similarly, virtualization techniques can be applied to other IT infrastructure layers - including networks, storage, laptop or server hardware, operating systems and applications. This blend of virtualization technologies - or virtual infrastructure - provides a layer of abstraction between computing, storage and networking hardware, and the applications running on it. Figure 2.1 shows the elements and structure of operating system before virtualization and figure 2.2 shows after virtualization. The deployment of virtual infrastructure is non-disruptive, since the user experiences are largely unchanged. However, virtual infrastructure gives administrators the advantage of managing pooled resources across the enterprise, allowing IT managers to be more responsive to dynamic organizational needs and to better leverage infrastructure investments.

Virtualization technology allows data-centers to lease virtual machines to end users, rather than traditional dedicated machines. This system is the basis of Cloud Computing's on- demand, utility model. Virtualization can also increase the efficiency of existing machine rooms, reducing the number of physical servers required through consolidation of existing applications by introducing multiple virtual machines per server and thereby increasing resource utilization. It is assumed that this increases efficiency; it allows you to do more work with less IT equipment.

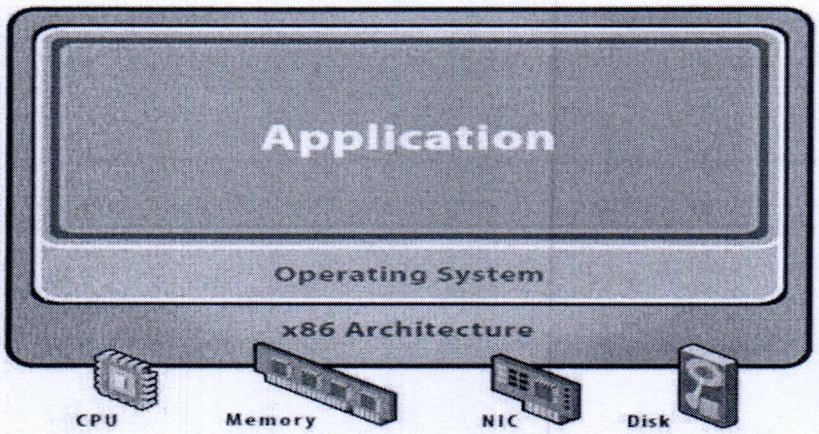


Figure 2.1: Before virtualization [1]

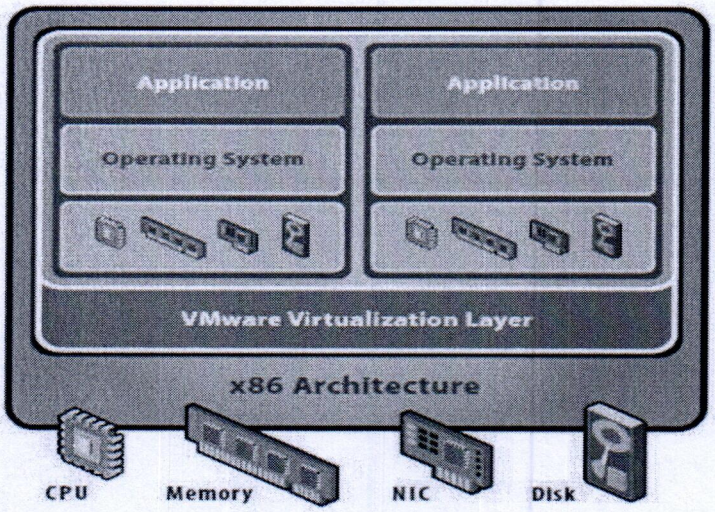


Figure 2.2: After Virtualization [1]

In the 1960's, when the data center technicians had problems with traditional data centers (static data centers), the dynamic solution was the solution to their problems. The major problem was very simple, if they had high load on their data center occasionally, they could not manage it easily, hence the greatest achievement was the new technology called as server virtualization [2].

It was a long way from completely static data centers to dynamic data centers that can provide dynamic resource allocation for IT environment providers. They designed the software which creates different isolated boxes and shares all of the hardware resources

between them and give them resources as they need. These boxes as they called virtual machines (VM), acts and services the clients like real servers with their strength and their weakness. The server virtualization conceptually designed for data centers that they can dynamically control and share all available resources over their data centers and it could be increased or decreased on-demand.

2.2. Virtualization in cloud

Virtualization is one of the major components of cloud computing that helps to emergence of cloud computing. To understand cloud computing it is important to understand the concepts like network virtualization or storage virtualization.

The main components of virtualization in the cloud are virtual machines, because all of the operating systems and applications are inside them. They are like a container which isolated and separated from each other, even in the same physical host. Based on the basic role of cloud as-pay-you-go, the vendors give you that ability to access these provided virtual machines and in some cases they will make these virtual machines like an actual computer and you can purchase them for a limited time and take advantage of this ability without any worries about how they operate. That thing you are purchasing is the availability of these services [2]. Vendors in exchange promise to give you these services without any interrupt that make the availability in high level. Virtualization provides some major advantages like:

A) High availability of resources in cloud

With the help of virtualization we achieve the concept of high availability of resources in cloud. As we know that the entire machine here is virtual, it means they are pieces of software that easily can deploy them whenever and anywhere we want. So if any issue occurs with any physical servers or related devices, we can easily move the virtual machines to a new location. By using the concept of virtualization we can easily reduce or avoid the downtime of any datacenters [3].

B) Scalability of cloud

Scalability of the cloud is also possible with the help of virtualization. As scalability is one of the important features of the cloud which insure that cloud will provide the services even if customer demand increases suddenly by adding number of virtual machines. The expansion and shrinking of cloud to meet the customer demands comes under dynamic virtualization [2].

C) Storage Virtualization

As the size of the data is increasing day by day storage of the data is one of the major issues. Physical storage is not directly connected to the servers. They are connected through the network. One of the most popular protocols use for storage virtualization is a storage area network (SAN) and network attached storage (NAS). For faster interaction, fiber cables are used. The significant point of the storage virtualization is to hide geographical positions of the data over the cloud environment. In storage virtualization still the management of the data which can be distributed among the network is the one of the important issues in cloud storage [4].

2.3 Load Balancing in Cloud

Load Balancing (LB) is the method of dissemination of workload among nodes in an appropriate way with the end goal that it guarantees no node in the framework is over-burden or sits idle. Cloud LB diminishes costs and augments resource accessibility. Scheduling or allocation of user request in cloud environment is an NP – Hard optimization problem. In cloud framework there is some load assigned to each node or servers. This assigned load on the node might be overload, under-loaded or equally loaded. Situation like underloaded and overloaded causes different system failures results in power failure, increase response time and machine failure. Therefore load balancing is required to overcome all these failures. Load on Virtual machines can be categories as Memory load, Computation (CPU) load and network load etc.

Resource allocation in the cloud is based on the current state of the VM. In allocation policy where the current load information of VMs is available before the allocation is said to be a dynamic strategy. In strategic allocation there in no need of VMs prior load

information. In static load balancing, run time VM load information gathering is not possible. Figure 2.3 shows classification of load balancing algorithms.

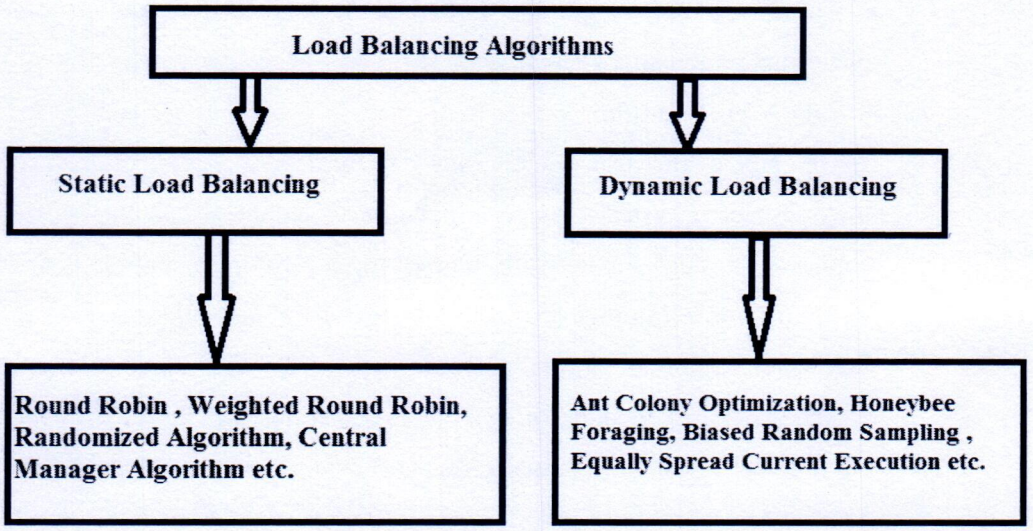

Load Balancing Algorithms
Static Load Balancing
Dynamic Load Balancing
Round Robin , Weighted Round Robin, Randomized Algorithm, Central Manager Algorithm etc.
Ant Colony Optimization, Honeybee Foraging, Biased Random Sampling , Equally Spread Current Execution etc.

Figure2.3. Classification of Load Balancing Algorithm

2.3.1 Static Load Balancing Algorithms

Static Algorithms are made for those systems which have very low variations in load. In Static Algorithm, the entire traffic is equally divided amongst the servers. This algorithm needs an in-depth knowledge of server resources for better performance of the processors which is determined at the beginning of the implementation.

However, the decision of load shifting does not depend on the present state of the system. There is one major drawback of Static Load Balancing Algorithm that is load balancing tasks work only after they are created, it could not be implemented to other devices for load balancing [3]. Some of the static load balancing algorithms is Round Robin, Weighted Round Robin, Randomized Algorithm etc.

A) **Round Robin Algorithm**: This is amongst the easiest methods of load balancing. As the name suggests, Round Robin Load Balancing Algorithm uses the round-

robin method for assigning the jobs[5]. Firstly, it chooses the first node on a random basis and assigns the tasks to the other nodes in round robin way. Processors circularly assign each process without defining any priority. This results in a faster response in case of similar workload distribution among the processes. All the processes have different loading time. Therefore, some nodes might be heavily loaded, while the others may remain under-utilized.

B) **Weighted Round Robin Algorithm**: A Weighted Round Robin Load Balancing Algorithm is developed to enhance the most challenging issues of Round Robin Algorithm. In this algorithm, there is a designated prescribed weight and jobs which are distributed as per the values of the weight. Those processors that have greater abilities are assigned a bigger value. Therefore, highest weighted servers will get more tasks. When the entire weights come in level, servers will get steady traffic.

C) **Randomized Algorithm**: In this algorithm the node is selected on a random way, without having any information about the current or previous load on the node. Each node maintains its own load record. The computing nodes are selected randomly to balance the load. This algorithm distributes the load randomly by first checking the size of the process and then transferring the load to a VM which is lightly loaded.

D) **Central Manager Algorithm**: In this algorithm, a central processor selects the host for new process. The minimally loaded processor depending on the overall load is selected when process is created. Load manager selects hosts for new processes so that the processor load confirms to same level as much as possible. From the on hand information on the system load state central load manager makes the load balancing judgment. This information is updated by remote processors, which send a message each time the load on them changes.

E) **Min- Min Algorithm**: This algorithm begins with a set of all unassigned jobs. Minimum completion time for all jobs is calculated and the job with minimum completion time is selected. Then, the node which has the minimum completion time for all jobs is selected. Finally, the selected node and the selected job are

mapped. This process is repeated until all the unassigned jobs are assigned. In Min-Min algorithm, a request with long processing time may wait for a long time.

F) **Max-Min Algorithm**: Max-Min algorithm calculates the minimum completion time for all jobs. After finding out minimum completion time of jobs, the maximum value is selected. The machine that has the minimum completion time for all the jobs is selected. Finally the selected node and the selected job are mapped.

2.3.2. **Dynamic Load Balancing Algorithms**

Dynamic algorithm are more flexible than the static algorithm and they doesn't rely on prior knowledge but depends on current state of the system. In a distributed system [5], dynamic load balancing has two different ways: distributed and non-distributed. In the distributed one, this algorithm is executed by all nodes present in the system and the task of load balancing is shared among these servers. The interaction among nodes to achieve load balancing can take two forms: cooperative and non-cooperative. In the first one, the nodes works side-by-side to achieve a common objective which means is to improve the overall response time, etc. In the second form, each node works independently toward a goal local to it [6]. Some of the dynamic load balancing algorithms is Ant Colony Optimization, Honeybee Foraging, Biased Random Sampling, Equally Spread Current Execution etc.

A) **Ant Colony Optimization**: Aim of the ant colony optimization to search an optimal path between the source of food and colony of ant on the basis of their behavior. This approach aims efficient distribution of work load among the node. When request is initialized the ant starts movement towards the source of food from the head node. Regional Load Balancing Node (RLBN) is chosen in Cloud Computing Service Provider (CCSP) as a head node. Ants keep records the every node they visits ant record their data for future decision making .Ant deposits the pheromones during their movement for other ants to select next node The intensity of pheromones can vary on the bases of certain factors like distance of food, quality of food etc. When the job gets successful the pheromones is updated. Each ant build their own individual result set and it is later on built into a

complete solution. The ant continuously updates a single result set rather than updating their own result set. By the ant pheromones trials, the solution set is continuously updated.

B) **Honeybee Foraging**: It is a nature inspired decentralized load balancing technique which helps to achieve load balancing across heterogeneous virtual machine of cloud computing environment through local server action and maximize the throughput [7]. The current workload of the VM is calculated then it decides the VM states whether it is over loaded ,under loaded or balanced .according to the current load of VM they are grouped. The priority of the task is taken into consideration after removed from the overload VM which are waiting for the VM .Then the task is schedule to the lightly loaded VM. The earlier removed task are helpful for the finding the lightly loaded VM. These tasks are known as scout bee in the next step. Honey Bee Behavior inspired Load Balancing technique reduces the response time of VM and also reduces the waiting time of task.

C) **Biased Random Sampling**: Biased Random Sampling Load Balancing Algorithm is dynamic approach; the network is represented in the form of virtual graph. Each server is taken as a vertex of the node and the in degree represents the available free resources the nodes have. On the basis of the in degree the load balancer allocates the job to the node. The nodes have at least one in degree then load balancer allocates the job to that node. When the job is allocates to the node then the in degree is decrement by one, and it's get incremented again when job gets executed. Random sampling technique is used in the addition and deletion of the processes. The processes are centralized by the threshold value, which indicates the maximum traversal from one node to destination node. The length of traversal is known as walk length. The neighbor node of the current node is selected for the traversal. After receiving the request, load balancer selects a node randomly and compares the current walk length with the threshold value. If the current walk length is equal to or greater than the threshold value, the job is executed at that node. Otherwise, the walk length of the job is incremented and

another neighbor node is selected randomly. The performance is decrease as the number of servers increase.

D) **Equally Spread Current Execution:** The Load balancer makes an attempt to safeguard equivalent load to any or all the VM's connected with the data centers (DC). It appropriates the heap to VM by checking the heap at current time and trade of the heap to that VM which is less stacked and result in less time taken, and gives high throughput. This Load balancer keeps up a record table of VM's and furthermore number of solicitations starting at now distributed to the VM . Exactly when all the VM's are stacked and if the request originates from the DC to assign the new VM, it checks the run down table for least stacked VM. The DC gives the request to the VM perceived. Right when VM completes the errand, it is educated to DC which is furthermore told by the load balancer.

2.4 Challenges of Load Balancing in cloud

1. Throughput- It is calculated as number of tasks executed in a specific time interval and it should by high in order to increase system performance.
2. Response Time- The amount of time taken by a load balancing algorithm to respond and it should be minimized.
3. Scalability- Ability to perform load balancing algorithms on n number of nodes with the same quality and this should be maximized.
4. Resource Utilization- Used to check the utilization of resources and it should be optimized for efficient load balancing.
5. Performance- Used to check the efficiency of the system i.e. maximum work done in minimum resources efficiently and at a faster rate and this factor should be maximized.
6. Overhead- Determines amount of overhead caused due to inter-process communication, movement of tasks which is required to implement load balancing algorithms and this should be minimized to work efficiently.
7. Fault Tolerance System- Ability to handle failure efficiently.

REFERENCES

[1] VMware Virtualization whitepaper "https://www.vmware.com/pdf/virtualization.pdf"

[2] Richard Scroggins, "Emerging Virtualization Technology" , Global Journal of Computer Science and Technology, Volume 17 Issue 3 Version 1.0 Year 2017, pp 1-7

[3] Reena Panwar, Bhawna Mallick "A Comparative Study of Load Balancing Algorithms in Cloud Computing", International Journal of Computer Applications, Volume 117 – No. 24, May 2015, pp 33-37

[4] Kunjal Garala , Namrata Goswami, Prashant D. Maheta, "A performance analysis of load Balancing algorithms in Cloud environment", International Conference on Computer Communication and Informatics (ICCCI), Jan 2015, Coimbatore, India.

[5] M. D. Dikaiakos, G. Pallis, D. Katsa, P. Mehra, and A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research", in Proc. of IEEE Journal of Internet Computing, Vol. 13, No. 5, pp. 10-13, 2009.

[6] Dharmesh Kashyap, Jaydeep Viradiya, "A Survey Of Various Load Balancing Algorithms In Cloud Computing", International Journal of Scientific and Technology Research, Vol 3, Issue 11,Nov 2014,pp 115-119

[7] T. Deepa, Dhanaraj Cheelu, "A comparative study of static and dynamic load balancing algorithms in cloud computing", International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Aug 2017, Chennai, India

[8] Sushil Kumar, Deepak Singh Rana, "Various Dynamic Load Balancing Algorithms in Cloud Environment: A Survey", International Journal of Computer Applications, Volume 129 – No.6, November 2015, pp 14-19.

[9] Surbhi Kapoor "A Survey on Dynamic Load Balancing algorithms in Cloud Computing", Advances in Computer Science and Information Technology , Volume 2, Number 7; April – June, 2015 pp 87 – 91.

Chapter 3
REVIEW OF LITERATURE

This chapter discusses about what are the issues related to designing load balancing algorithms in cloud environment.

3.1. Design Issues related to good Load Balancing Algorithms

3.1.1. Load estimation policy

It determines the total workload on the node in a system. The load can be estimated on the basis of total number of processors on the node, resource demands, architecture and speed of the nodes processor and instruction mixes of these processes. Several load-balancing algorithms use the total number of processes to achieve efficiency. In some cases the true load could vary widely depending on the remaining service time of processes on a node, which can be measured in several ways such as Memory less method, past repeats and Distribution method. An acceptable method for use as the load estimation policy in these systems would be to measure the CPU utilization of the nodes. It can be measured by setting up a timer to periodically check the CPU state (idle/busy).

3.1.2. Process transfer policy

It selects the processor for transferring the load which would increase the throughput and response time. The idea of using this policy is to transfer processes from heavily loaded nodes to lightly loaded nodes. Most of the algorithms use the threshold policy to decide whether the node is lightly-loaded or heavily-loaded. Threshold value is a limiting value of the workload of node which can be determined by:

Static policy: Predefined threshold value for each node depending on processing capability.

Dynamic policy: threshold value is calculated from average workload and a predefined constant.

Below threshold value node accepts processes to execute, above threshold value node tries to transfer processes to a lightly-loaded node. Single-threshold policy may lead to

unstable algorithm because under loaded node could turn to be overloaded right after a process migration. To reduce instability double-threshold policy has been proposed which is also known as high-low policy.

3.1.3. Location policy

The selection of destinations node for process execution is made by the location policy of a scheduling algorithm. The policies used can be:

Threshold method- This policy selects a random node and checks whether the node is able to receive the process then it transfers the process. If node rejects the transfer another node is selected randomly. This continues until probe limit is reached.

Shortest method- In these M distinct nodes is chosen at random and each is polled to determine its load. The process is transferred to the node having the minimum value unless its workload value prohibits it from accepting the process. To improve this it is necessary to discontinue probing whenever a node with zero loads is encountered.

Bidding method- In this method, nodes contain managers to send processes and contractors to receive processes. Managers broadcast a request for bid and contractors respond with bids where the manager selects the best offer. The winning contractor is notified and asked whether it accepts the process for execution or not. A contractor is never forced to accept remote process.

Pairing- The pairing policy is used to reduce the variance of load only between nodes of the system. Each node asks some randomly chosen node to form a pair with it. If it receives a rejection it randomly selects another node and tries to pair again. Two nodes that differ greatly in load are temporarily paired with each other and migration starts. The pair is broken as soon as the migration is over. A node only tries to find a partner if it has at least two processes.

3.1.4. State Information Exchange policy

It acquires workload related information about the system such as nature of workload and the average load on each node. Dynamic policies require frequent exchange of state information among the systems. These extra messages arise two opposite impacts one is

that it increasing the number of messages gives more accurate scheduling decision and the other is it increasing the number of messages raises the queuing time of messages. State information policies that can be used are:

Periodic broadcast-

Each node broadcasts its state information after the elapse of every T units of time. The problems caused are heavy traffic, poor scalability.

Broadcast when state changes-

This method avoids fruitless message exchange by broadcasting the state only when a process arrives or departures. Further improvement that can be done is to broadcast only when state switches to another region.

On-demand exchange-

In this method a node broadcasts a State-Information- Request message when its state switches from normal to either under-loaded or overloaded region. On receiving this message other nodes reply with their own state information to the requesting node. Further improvement can be that only those nodes reply which are useful to the requesting node and use of State Information Request.

Exchange by polling-

The polling method overcomes poor scalability that comes from broadcast messages. There is no need for a node to exchange its state information. The partner node is searched by polling the other nodes on by one until poll limit is reached.

3.1.5. Priority assignment policy

In a distributed operating system that support process migration, a priority assignment rule for scheduling the local and remote process of a node should be planned. One of the following priority assignment rules may be used:

1) Selfish priority assignment rule-

In this the local processes are given higher priority than remote processes. This rule has the worst response time performance of the three policies due to poor performing processes. Best response time performance is achieved for local processes.

2) Altruistic priority assignment rule-

Remote processes are given higher priority than local processes in this rule. It has the best response time performance of the three policies. Remote process incur lower delays than local processes as local processes are treated as principal while remote processes are secondary workload.

3) Intermediate priority assignment rule-

Priority in this rule is decided depending on number of local and remote processes on a node. When the number of local processes is greater or equal to the number of remote processes than local processes are given higher priority or else remote processes are given higher priority than local processes. Performance of this rule is between the two above policies. Overall response time performance is close to altruistic policy.

4) Migration limiting policy for Load-balancing algorithms-

This policy determines the total number of times a process can migrate. One of the following two policies may be used-

- Uncontrolled: A remote process arriving at a node is treated just as a process originating at a node due to which a process may be migrated any number of times.
- Controlled: This avoids the instability of the uncontrolled policy and uses a migration count parameter to fix a limit on the number of time a process that can migrate. A long process may be allowed to migrate more times compares to short process.

3.2. Literature Analysis

D. Chitra Devi and V. Rhymend Uthariaraj [2] discussed how to overcome issues related to round-robin load balancing algorithm. He talked about the weighted round-robin scheme in which total turnaround time for each user job is improved.

Sudha Senthil Kumar et al. [3] talked about the improved version of the Honey bee foraging Algorithm. They did the addition of adaptive neighboring search and site abandonment strategy with Honey bee foraging algorithm. They figure it out that efficiency is improved using this combined approach.

Carrera et al. [4] have explored the problem of managing heterogeneous workloads in a virtualized data centers. They have proposed a technique which dynamically modifies

the heterogeneous workload placement on a same physical hardware. The utility functions used in this technique helped them to compare the performance of different workloads. This technique maximizes the performance of heterogeneous workloads.

Xu et al. [5] have explored number of load balancing algorithms for virtual machine placement in cloud computing. They have also explored the number of challenges like overhead, performance etc. related to load balancing in virtual environment. As a future scope they have talked about the designing of self-adaptive dynamic load balancing algorithm.

Chen [6] proposed an algorithm called PA-LBIMM (user Priority Aware Load Balanced Improved Min Min) scheduling algorithm which takes the characteristic of min-min scheduling algorithm as foundation to minimize the completion time of all resources and improve the load balance factor. It divides the user submitted tasks into two categories, VIP user tasks and ordinary tasks such that VIP tasks are executed first and are assigned to only VIP resources. After all the VIP tasks are assigned, then ordinary tasks are scheduled to both VIP and ordinary resources. After all the tasks get assigned, load balancing is then done by selecting the task with minimum completion time from the most heavily loaded resource. Now the completion time of this task is calculated on all the other resources and a minimum value is obtained. If this minimum value of completion time is less than the make-span of the tasks, then the particular task is reassigned to that resource.

This continues until no other task on the most heavily loaded resource with the minimum completion time needs rescheduling. Since cloud is a pay according to use model, so issuing tasks on the grounds of user priorities is a good idea for scheduling the load.

A Load Balancing Ant Colony Optimization (LBACO) algorithm is proposed in [7] that will balance load in a cloud environment using Ant colony Optimization algorithm as the foundation. It reduces the make-span of a given tasks set and balances the entire system load. The algorithm initializes the pheromone of all VMs on the basis of number of processors, their capacity and the bandwidth factor and places the ants on VMs randomly. It then choses the next VM for an ant on the basis of probability which is computed as the ratio of pheromone of that machine to the ratio of pheromone of all

machines. The one with the maximum probability among neighboring VMs is chosen as the next VM by the ant. When an ant completes the tour, the pheromone is updated locally and if the solution obtained is current optimal solution, then global updation of pheromone is done. The algorithm continues as long as iterative condition satisfies. For each iteration, all the ants complete their tour. This algorithm is simulated in CloudSim and the results have been compared with FCFS and ACO on the basis of load imbalance and make-span of the tasks and it has been shown that the proposed approach performs better than these algorithms.

Ghafari [8] proposed a Bee-MMT algorithm (Artificial Bee Colony algorithm- Minimal Migration Time) that makes use of bee colony algorithm to detect over utilized hosts and then using MMT selection for VMs, it decides which VMs can be shifted from the over utilized hosts. It then also finds the underutilized hosts and shifts all VMs assigned to these hosts to the other suitable hosts and then moves these hosts to switch off mode to save energy. This approach also cannot entirely solve the trade-off between response time of the tasks and energy consumption. This proposed algorithm can get greater power consumption than other algorithms thus supporting green computing concept.

Wenhong [9] proposed the DAIRS (Dynamic and Integrated Resource Scheduling) algorithm. Unlike other traditional algorithms, this algorithm not only considers CPU load but also takes into account memory and network bandwidth load for calculating the load on a particular machine. Four types of queues are maintained by the algorithm. These are waiting queue containing requests that are not allotted immediately but are waiting, request queue containing the requests that are new, optimizing queue is for those tasks that need to be reallocated and delete queue keeps the tasks whose end time is pending. New tasks are accepted from the queues on the basis of priority of the queues where priority list is such that waiting queue is having the highest priority, then are the request queue, then optimal queue and then delete queue with the lowest priority. The allocation algorithm which is being used for assigning tasks from the queues to the servers firstly sorts the servers in increasing order of utilization and then it divides the utilization of physical servers into multiple intervals.

3.3. Research Objectives

1. Comparison of load balancing with re-balancing to analyze the best technique for saving migration time related to load transfer.
2. To prioritize cloud customer based on services demand and geographical location for better customer satisfaction.
3. To improve the resource utilization in a cloud system using better load balancing techniques.

REFERENCES

[1] J. Srinivas, K. Venkta Subha Ready, Dr. A. Moiz Qyser, " Cloud Computing Basics", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 1, Issue 5, July 2012, pp 343-346.

[2] D. Chitra Devi and V. Rhymend Uthariaraj, "Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks", The Scientific World Journal, Vol 2016, Article ID 3896065, 14 pages

[3] Sudha Senthil kumar, K.Brindha, Rathi, Angulakshmi, Jothi and Yash Vardhan Thirani (2017), —Honey-Bee Foraging Algorithm for Load Balancing in Cloud Computing Optimization", International Journal of Engineering Science and Computing, Vol. 7, Issue No. 12, pp. 15840-15844.

[4] D. Carrera, M. Steinder, I. Whalley, J. Torres and E. Ayguade, "Managing SLAs of heterogeneous workloads using dynamic application placement", HPDC '08 Proceedings of the 17th international symposium on High performance distributed computing, New York, USA, 2008.

[5] Minxian Xu Wenhong Tian, Rajkumar Buyya1, "A survey on load balancing algorithms for virtual machines placement in cloud computing" ,Wiley, Feb 2017, pp. 1-16

[6] Huankai Chen; Wang, F.; Helian, N.; Akanmu, G., "Userpriority guided Min-Minscheduling algorithm for load balancing in cloud computing," Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference on , vol., no., pp.1,8, 21-23 Feb. 2013

[7] Kun Li; Gaochao Xu; Guangyu Zhao; Yushuang Dong; Wang, D., "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization," Chinagrid Conference (ChinaGrid), 2011 Sixth Annual , vol., no., pp.3,9, 22-23 Aug. 2011.

[8] Ghafari, S.M.; Fazeli, M.; Patooghy, A.; Rikhtechi, L., "Bee- MMT: A load balancing method for power consumption management in cloud computing,"

Contemporary Computing (IC3), 2013 Sixth International Conference on , vol., no., pp.76,80, 8-10 Aug. 2013

[9] Wenhong Tian; Yong Zhao; Yuanliang Zhong; Minxian Xu; Chen Jing, "A dynamic and integrated load-balancing scheduling algorithm for Cloud datacenters," Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on , vol., no., pp.311,315, 15-17 Sept. 2011

CHAPTER 4

LOAD RE-BALANCING APPROACH FOR DISTRIBUTED LOAD IN CLOUD ENVIRONMENT

4.1 INTRODUCTION

In this section, load re-balancing approach was outlined in context to cloud load balancing issue. Hadoop File System was developed using distributed file system design. It runs on commodity hardware. Unlike other distributed systems, HDFS is highly fault tolerant and designed using low-cost hardware. Offloading the load rebalancing task is suggested to have storage nodes by having the storage nodes balance their loads spontaneously. The data, center operators in the background, virtualizes the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple servers. Data robustness is a major requirement for storage systems. There have been many proposals of storing data over storage servers. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message. A decentralized erasure code is suitable for use in a distributed storage system.

4.2 LOAD BALANCING

Load balancing is dividing the amount of work that a computer has to do between two or more computers so that more work gets done in the same amount of time and, in general, all users get served faster. For example a web application or websites can be accessed by a plenty of users at any point of time. It is very difficult for the applications to manage all these users at one time and provide response frequently [1][2]. For a website owner it creates a feeling that his potential customer will go down because of not getting the response in a stipulated time. Here we have this website be hosted on cloud computing environment. This type of problem is very common in distributed

environment [3][5][6]. To overcome this type of problem we have the concept of load balancing.

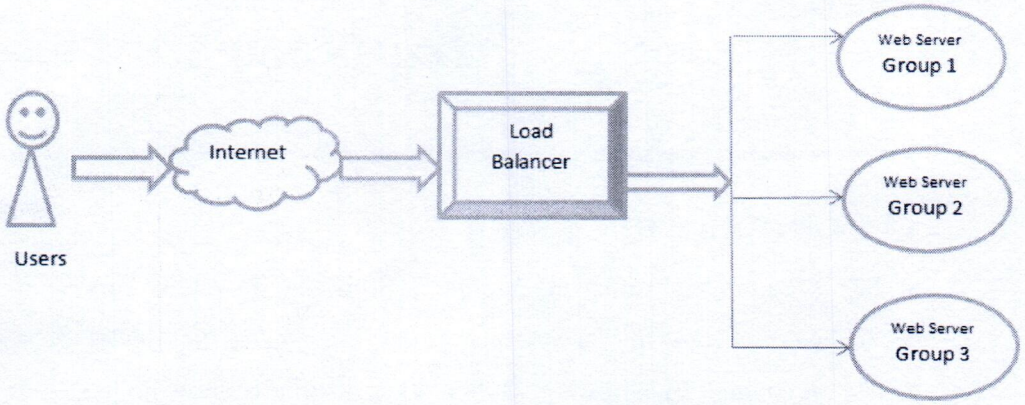


Figure 4.1: Radio Energy Dissipation Model

Simple load balancing diagram is shown in figure 4.1. The main objective of using load balancing is:

1. To maintain system durability.
2. To improve system performance.
3. To protect against system failure.

For balancing the load we have number of load balancing algorithms [4][7]. We categorize these algorithms basically in two types:

A) **Static load balancing**: Static load balancing is used for homogeneous resources. In this the load balancer has the initial information about the servers and virtual machines and load is distributed on the basis of initial information.

B) **Dynamic Load balancing**: Dynamic load balancing is used for heterogeneous resources. Here in this system the load balancer maintain all the information's about the servers /VMs state and load on each server and VMs. This information related to VMs is updated periodically. It is more stable as compared to static because the load balancer has the current information about the VM (or resources).

4.3 LOAD REBALANCING APPROACH

Assumptions:

For implementation of our approach we have the following assumptions:

1. Initially we have 5 Virtual Machines (Vm's)or Servers having same CPU and storage capacity. These VM's are numbered from 1 to 5 (like VM_1, VM_2, VM_3, VM_4, VM_5).
2. We have 3 users (U1, U2, and U3) with some load in terms of files or processes (like P1, P2, P3, P4, and P5).These load (file or processes also called chunks) are of equal size. Number of chunks with each user is shown in table 4.1.
3. All the virtual machines have zero loads initially.
4. For balanced condition each VM's or Server need to process **A** chunks.

$$A = \frac{\sum_{p \in P} |P_f|}{n}$$

Where: P_f is fixed size process

P is total number of process

n is total number of VM's

S.No	User	Processes
1	U1	P1,P2,P3
2	U2	P1,P2,P3,P4,P5
3	U3	P1,P2,P3,P4,P5
4	U4	P1,P2

Table 4.1 Initial User with their loads.

VM1	VM2	VM3	VM4	VM5
U1-P1	U1-P2	U1-P3		
U2-P1	U2-P2	U2-P3	U2-P4	U2-P5
U3-P1	U3-P2	U3-P3	U3-P4	U3-P5
U4-P1	U4-P2			
4(processes)	4(processes)	3(processes)	2(processes)	2(processes)

Table 4.2 Initial Load distribution when VM's are idle.

VM1	VM2	VM3	VM4	VM5
U1-P1	U1-P2	U1-P3	U4-P1	U4-P2
U2-P1	U2-P2	U2-P3	U2-P4	U2-P5
U3-P1	U3-P2	U3-P3	U3-P4	U3-P5
3(processes)	3(processes)	3(processes)	3(processes)	3(processes)

Table 4.3: Load distribution on VM's after rebalancing

We will focus on designing a load rebalancing algorithm to reallocate the chunks to achieve the uniform load distribution to the system as much as possible. We will also try to reduce the movement cost i.e. the migration of chunks caused for balancing loads of servers (or VM's).

Steps to be followed in our algorithms

1. First load is assigned to these machines randomly using the concepts of static load balancing as shown in Table 4.2.Each VM's get some chunks to process.
2. Now for a load balanced condition each VM or Server needs to process **A** chunks. Each chunk server node **I** first estimate whether it is under loaded (light) or overloaded (heavy) without global knowledge.
3. A VM is under loaded if, number of processes< Minimum threshold value of VM (ΔTmin) and a node is overloaded if, Number of processes >Maximum threshold value of VM (ΔTmax).
4. In our approach we discussion, if a Vm **i** departs and rejoins as a successor of another Vm **j**, then we represent Vm **i** as Vm **j+1**, Vm j 's original successor as Vm **j+2**, the successor of Vm j 's original successor as Vm j+3, and so on. For each Vm I $\in$ V, if Vm **i** is light, then it seeks a heavy node and takes over at most A process from the heavy Vm.
5. Table 4.3 shows the resulted load on each VMs or servers after the application of load re-balancing approach.

4.4 RESULT AND DISCUSSION

We had used the cloudsim and cloud analyst for the simulation purpose. Cloud sim is generally used for simulation of load balancing problems. We are using simulation tools and virtual machines environment because installation and preparation of distributed environment is costing a lot for testing purpose.

In cloud analyst we first configure the simulation environment. In that we first design the data centers having number of VMs with their capacity. We also design the user base with their request size (that is to be measured in terms of file size or processes size).

After that we selected load balancing policy as round robin. Once everything is set we run the simulation and get the results shown in fig 3.2.

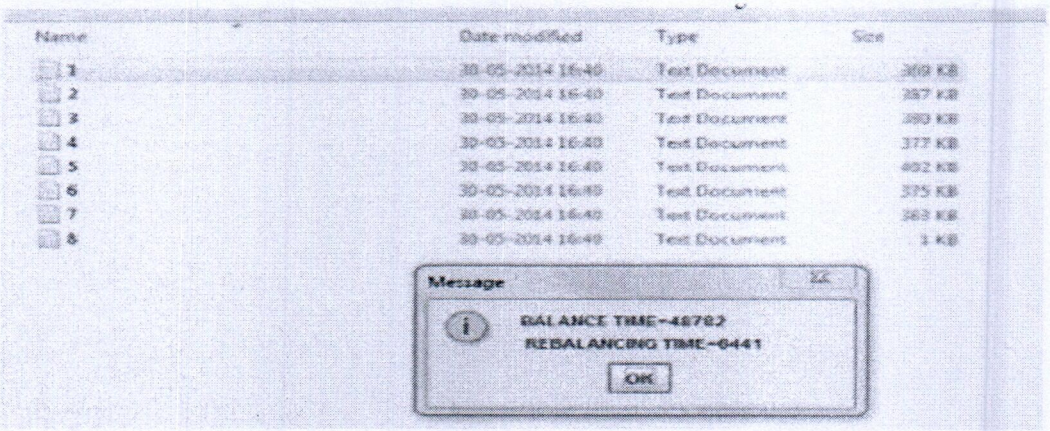


Figure 4.2 Simulation Results

From fig 3.2 we concluded that rebalancing processes will divide the workload equally on each VM or Data Centre. The size of the file is of approximately 380KB.We have also seen that it takes around 48782 milliseconds for load balancing and 6441 milliseconds for rebalancing. From fig 3 we also concluded that rebalancing time is far more less than balancing.

4.5 CONCLUSION

Load balancing is one of the important challenges in front of cloud service providers. If there is a proper load balancing algorithm is in place response time to customer is going

to be reduced. Balancing the load by using rebalancing algorithms is also very effective for distributed systems. In this paper we have shown rebalancing approach using simulated environment. There is also too much scope of improvement in rebalancing in terms of exact load rebalancing with less time of load migration.

REFERENCES

[1] H.-C. Hsiao, H. Liao, S.-S. Chen, and K.-C. Huang, "Load Balance with Imperfect Information in Structured Peer-to-Peer Systems,"IEEE Trans. Parallel Distributed Systems, vol. 22, no. 4, pp. 634-649, Apr. 2011.

[2] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica, "Load Balancing in Structured P2P Systems," Proc. Second Int'l Workshop Peer-to-Peer Systems (IPTPS'02), pp. 68-79, Feb. 2003.

[3] Hung-Chang Hsiao, Hsueh-Yi Chung, Haiying Shen, Yu-Chang Chao," Load Rebalancing for Distributed File Systems in Clouds", IEEE transactions on parallel and distributed systems, vol. 24, no. 5, may 2013.

[4] SmitaSalunkhe, S. S. Sannakki, "Load Balancing in Clouds," International Journal of Emerging Trends in Electrical and Electronics, vol. 11(2), pp. 126-131, June 2015.

[5] Y. Zhu and Y. Hu, "Efficient, Proximity-Aware Load Balancing for DHT-Based P2P Systems," IEEE Trans. Parallel and Distributed Systems, vol. 16(4), pp. 349-361, Apr. 2005.

[6] K. S. Sundaram, "Load Balancing for Distributed File System," International Journal of Advanced Computational Engineering and Networking, ISSN Vol. 1(1), pp. 25-30, Mar. 2013

[7] Suriya Begum and Dr. Prashanth, "Investigational Study of 7 Effective Schemes of Load Balancing in Cloud Computing," International Journal of Computer Science Issues, Vol. 10(1), pp. 276-287, Nov. 2013.

[8] B. Godfrey, K. Lakshminarayanan, R. Karp and I. Stoica S. Surana, "Load Balancing in Dynamic Structured P2P Systems," Performance Evaluation, vol. 63(6), pp. 217-240, 2006 Mar.

[9] S.Penmatsa and T.Chronopoulos, Game-theoretic static load balancing for distributed systems, Journal of Parallel and Distributed Computing, vol.71, no.4, pp.537-555, Apr. 2011.

CHAPTER 5

LOAD BALANCING IN CLOUD USINGPRIORITIZATION BASED ON QUALITY OF SERVICES (QOS) DEMAND

5.1 INTRODUCTION

Cloud computing is an exceptionally current theme and the term has picked up a great deal of consideration as of late. It can be characterized as on request pay according to utilize display in which shared assets, data, programming and different gadgets are given by the customers' prerequisite when required. Human reliance on cloud is clear from the way that today's most well-known social organizing, email, archive sharing and web based gaming destinations are facilitated on cloud. Google, Microsoft, IBM, Amazon, Yahoo and Apple among others are exceptionally dynamic in this field [1].

Load balancing is the significant zone of worry in cloud computing. It is a component that disperses the workload equitably over every one of the hubs in the entire cloud to maintain a strategic distance from a circumstance where a few hubs are vigorously stacked while others are sit still or doing little work. Load balancing is the way toward reassigning the aggregate burdens to the individual hubs of the aggregate framework to make the best reaction time and furthermore great usage of the assets. Cloud computing is an administration giving office over the web in which the heap adjusting is the one of the testing undertaking. Different strategies are to be utilized to improve a framework by distributing the loads to the hubs in an adjusting way however because of system blockage, transmission capacity use and so forth, there were issues are happened [1][2]. These issues were tackled by a portion of the current algorithms. A load balancing algorithm which is alterable in nature does not require any past data conduct of the framework, that is, it relies on upon the present conduct of the framework. There are different objectives that identified with the load balancing, for example, to enhance the execution significantly, to keep up the framework steadiness and so on. Contingent upon

the present condition of the framework, load balancing calculations can be arranged into two sorts they are static and dynamic algorithms.

5.2 LOAD BALANCING

It disseminates the aggregate load to the individual machines of the framework in such a way that each and every hub successfully uses the assets and limits the reaction time. It tackles situations where a portion of the hubs are over stacked or under stacked. A dynamic calculation does not consider the past state & relies upon the present conduct of the framework [3].The key qualities utilized are: execution of framework, correlation of load, security of various framework, collaboration between the hubs, estimation of load, choosing of hubs, nature of work to be exchanged and numerous different ones .Load balancing can be categories into basic two types:

A. Static load balancing:

Here the cloud supplier uses homogeneous assets. Here progressed provisioning is done in which the suppliers set up the suitable assets before beginning of administrations as per the agreements done. Non versatile assets are used due to the static condition. The cloud requires prior information of hubs limit, preparing power, memory, execution and insights of client necessities which are not variable [4].

B. Dynamic load balancing:

Heterogeneous assets are supplied by cloud here. Dynamic provisioning is done with adaptable assets in which the specialist co-op apportions more assets as they are expected to the client and expelled them when they are not utilized. Cloud does not require any previous learning [5] and the prerequisites of the clients are versatile. Dynamic condition is hard to be reproduced yet is very customizable with distributed computing condition.

5.3 PRIORITIZATION METHODOLOGY

It uses the priority which is assigned to both user tasks and datacenters depending on parameters stated.

A*) Prioritization on user:* After submitting the tasks, it will be processed by LB which calculates the priority. The user task will provide parameters like cost based on QoS request (CBQR) provided 60% priority which satisfies SLA conditions which aims to

increase the overall efficiency [3][8]. More the amount of CBQR, high the priority it will be assigned. Second parameter will be the count of VMs required which is assigned 40% priority. Less priority will be assigned to the task which will require more number of VM. Now the load balancer will calculate the total priority of the task using table 5.1 and 4.2.

B) *Prioritization on datacenters:* Again the prioritization to datacenters will be assigned by the load balancers on the basis of one parameter. More number of VMs a datacenter possess, higher priority it will be assigned. The table of the datacenters priority (table 5.3) will be available to load balancer [4]. On the basis of this the load balancer will prepare table of datacenters on the basis of their geographical location (GL) and the count of VMs they have and also the number of available VMs in a datacenter.

C) *Load balancer:* It is the middleware which is responsible for assigning the priority to user tasks and datacenters. It is the heart of the algorithm & contains table on CBQR which describes the range of priority.

CBQR		
Quality	Cost Range	Priority
Excellent	CBQR>1000000	60%
Good	50000<CBQR<=1000000	30%
Average	10000<CBQR<=50000	10%

Table 5.1: Priority assignment based on CBQR

Virtual Machine count(VMC)	
Total VMs required	Priority
1-3	40%
4-6	30%
7-10	20%
>10	10%

Table 5.2: Priority assignment based on VMs

Datacenter	
Total VMs	Priority
1-3	10%
4-6	20%
7-10	30%
>10	40%

Table 5.3: Priority assignment to Datacenter.

Priority table of user tasks				
UID	VMC	CBQR	Priority	GL
U1	6	1000001	90%	Asia
U2	8	60000	60%	Africa
U3	4	40000	30%	America
U4	7	20000	40%	Asia

Table 5.4: Priority table of users

Data Center in Asia Region			
DC ID	VMC	VMC available	Priority
DC 1	10	10	40%
DC2	8	8	30%
Dc3	7	7	30%

Table 5.5: Data center in Asia Region

Data Center in Africa Region			
DC ID	VMC	VMC available	Priority
DC4	9	9	30%
DC5	7	7	30%

Table 5.6: Data center in Africa Region

Data Centre in America Region			
DC ID	VMC	VMC available	Priority
DC6	4	4	20%
DC7	5	5	20%
Dc8	2	2	10%

Table 5.7: Data center in America Region

It also allocates priority to a parameter that is VMs count required. More the number of VM required less will be the priority as depicted in table 5.2. Also the load balancer will manage a table of priority of datacenters. The priority will be assigned to datacenters on the basis of VMC. More the VMC a datacenter had more is the priority. This is because higher VMC will lead in increase in overall efficiency of that datacenter and thus more priority.

D) *Priority calculation:* Here we have considered 4 users U1, U2, U3 and U4 whose parameters are stated in table 5.3. The load balancer will calculate the priority of each user task by using table 5.1 and 5. 2.

Priority is calculated as follows. Let us consider U1. VMC required here is 6 and by table 5.2 we can see that the priority for VMC=6 is 30%. Again CBQR is 1000001 and by looking in table 5.1 we can see that when CBQR>1000000 priority is 60%. Hence total priority to U1 is 90%. Similarly, priorities of all users can be calculated as demonstrated in table 5.3.

E) *Datacenter allocation process*: The LB now has table of user tasks & datacenter table along with their priority and GL, VMC, priority. The LB will look for the maximum priority task [6][7]. Next it will check mapping. It checks the GL of the task and searches for the cluster of datacenter of that location or nearby location in order to minimize response time. Load balancer will check data centers according to range of priorities of table 5.2. Once it founds a datacenter with VMs which satisfies the task need, it will check number of VMs available at that time. If count of VMs present are equal to VMs needed, task will be allocated to that VM otherwise another datacenter is searched. In our example the LB will first process task of U1. It will check its GL which is Asia. Then it will search for the cluster of datacenter that have GL= Asia or nearby Asia. Now the LB will check data centers according to their range of priorities & will apply best fit to the task [9][10].

F) *Waiting list*

1. Task left idle will be placed in the waiting queue according to their GL and decreasing priorities [5].
2. This waiting list is handled by the load balancer.
3. When execution gets over, it de-allocates the VMs and the status is updated in the DC table of that particular GL.
4. Request with higher priority in the waiting list is checked & executed else next task in the waiting list is considered.

5.4 PROPOSED ALGORITHM

Phase 1:

In first phase all VMs in the datacenter are available.

Step 1: Let there be N set of users {U1, U2, Un}.

Step 2: The load balancer will maintain 3 priority assignment tables that is table 5.1, 5.2 and 5.3.

Step 3: Load balancer receives task.

Step 4: Load balancer will calculate the priority of the user tasks using table 5.1 and 5.2.

Step 5: LB will maintain a table of datacenters by making cluster on the basis of their GL.

Step 6: The LB will be possessing information about the VMCs in the datacenter and the geographical location of that datacenter.

Step 7: Load balancer will pick up the task with high priority and will check its GL.

Step 8: Load balancer will look for cluster of datacenter with that particular GL.

Step 9: After finding cluster of datacenters with that particular GL load will be assigned to datacenter with the minimum number of VMs by applying best fit.

Step 10: Update the status of virtual machines available in the datacenters table (table 5.5, 5.6, 5.7).

Step 11: Begin the execution.

Step 12: De-allocates the VM once the execution is completed and update the status in datacenter table.

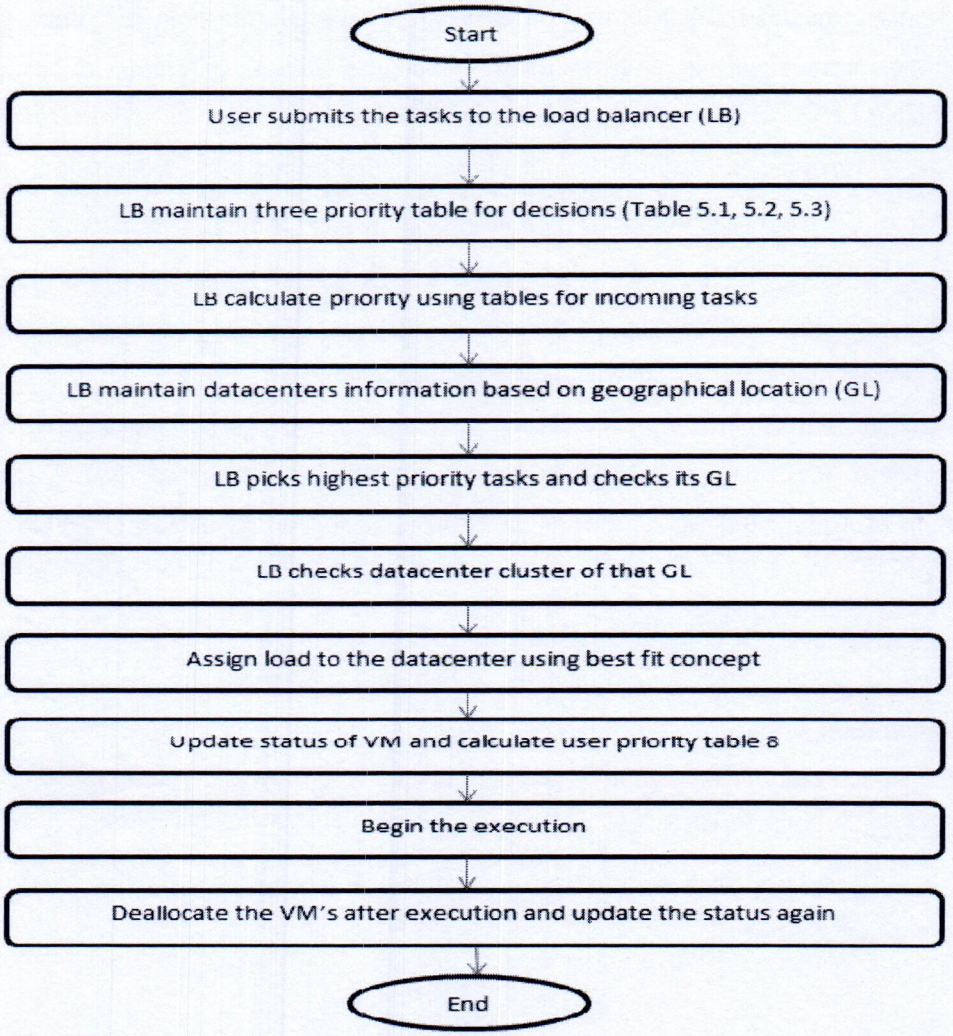

Start
User submits the tasks to the load balancer (LB)
LB maintain three priority table for decisions (Table 5.1, 5.2, 5.3)
LB calculate priority using tables for incoming tasks
LB maintain datacenters information based on geographical location (GL)
LB picks highest priority tasks and checks its GL
LB checks datacenter cluster of that GL
Assign load to the datacenter using best fit concept
Update status of VM and calculate user priority table 8
Begin the execution
Deallocate the VM's after execution and update the status again
End

Figure 5.1: Flowchart of phase 1.

Phase 2:

VMs in the datacenter are not available.

Step 1: Let there be N set of users {U1, U2… Un}.

Step 2: The load balancer will maintain 3 priority assignment tables i.e. table 5.1, 5.2 and 5.3.

Step 3: Load balancer receives task.

Step 4: Load balancer will calculate the priority of the user tasks using table 5.1 and 5.2.

Step 5: LB maintains table of datacenters by making cluster depending on GL and will update VMC.

Step 6: The load balancer will be possessing information about the Virtual Machine Counts in the datacenter, available virtual machines and the geographical location of that datacenter.

Step 7: Load balancer will pick up the task with high priority and will check its GL.

Step 8: Load balancer will look for cluster of datacenter with that particular GL.

Step 9: After finding cluster of datacenters with that particular GL, load balancer will look for datacenter with the nearest Virtual Machine Count required and will also check the available Virtual Machine Counts with datacenter.

Step 10: If it satisfies the need of that task, it is allocated to the datacenter. Else repeat step 9 to 10.

Step 11: Update the status of virtual machines available in the datacenters table (table 5.5, 5.6, and 5.7).

Step 12: Begin the execution.

Step 13: De-allocates the VM once the execution is completed and again update the status in datacenter table.

Step 14: If there is any task which is not allocated then it will be put in the waiting queue according to the decreasing order of priority.

Step 15: Once virtual machines are available tasks are picked up from waiting queue according to priority.

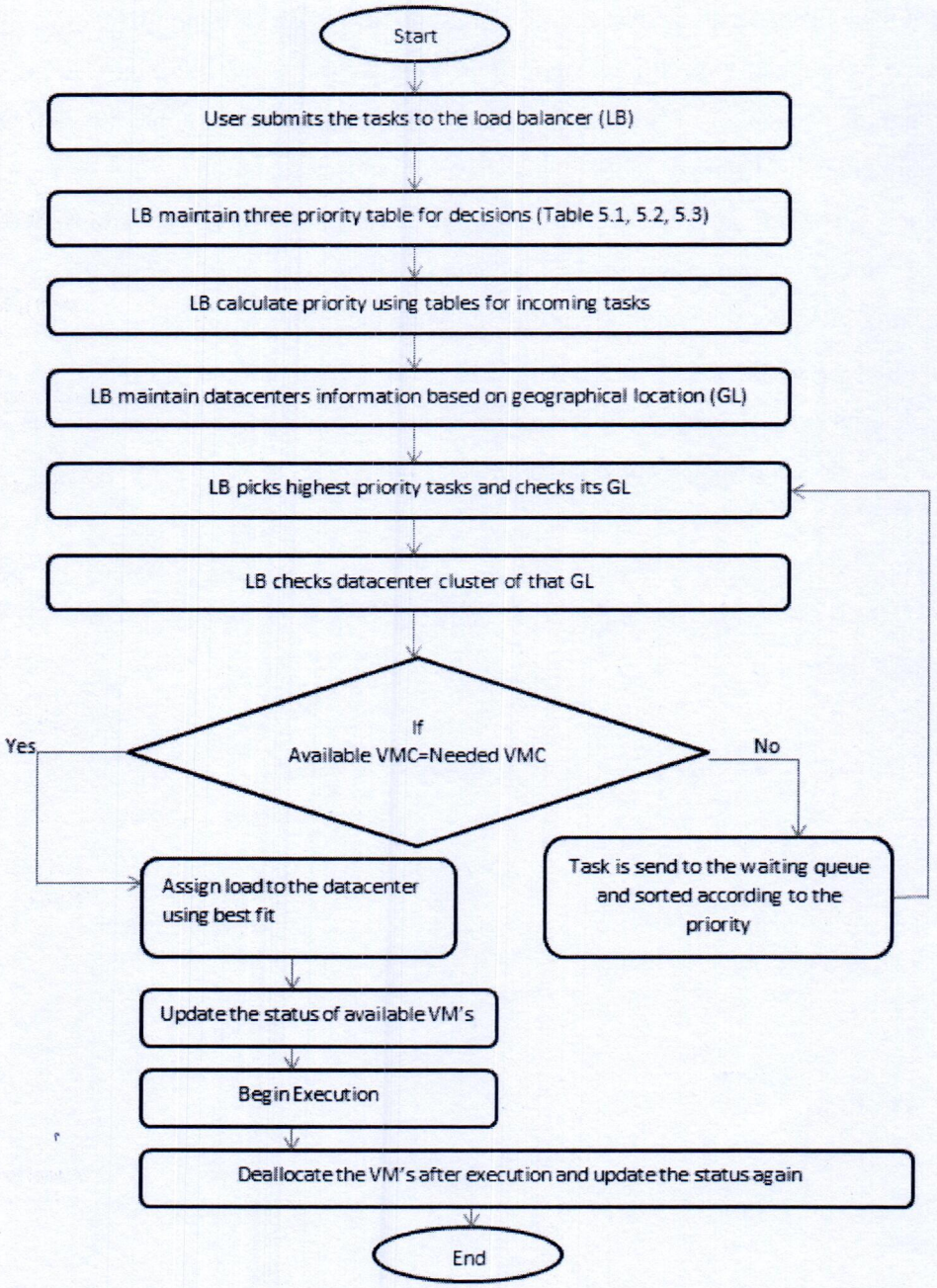

Start
User submits the tasks to the load balancer (LB)
LB maintain three priority table for decisions (Table 5.1, 5.2, 5.3)
LB calculate priority using tables for incoming tasks
LB maintain datacenters information based on geographical location (GL)
LB picks highest priority tasks and checks its GL
LB checks datacenter cluster of that GL
If
Available VMC=Needed VMC
Yes
No
Assign load to the datacenter using best fit
Task is send to the waiting queue and sorted according to the priority
Update the status of available VM's
Begin Execution
Deallocate the VM's after execution and update the status again
End

Figure 5.2: Flow chart for phase 2

5.5 RESULTS AND DISCUSSION

We have taken cloud analyst tool to represent our results. For simulation three user base (UB) were taken with different priorities. First we put the data center (DC) in the same geographical region of the user base and compute average response time. Next we put data center in different geographical location and compute again the average response time. We find out that the response time for the UB served from DC in same geographical location is far less as compared to UB served from DC in different locations. As per the results we try to minimize the response time [14]. Average response time is reduced if the cloud users will be served from same geographical region and of high priority.UB1 get cloud access first then UB2 and UB3as shown in table 5.8.Figure 5.3 variation in response time based on values of table 5.8.

S.No	User Base(UB) with priority	Average response time(ms)	
		UB served from Data Centre in same Geographical location	UB served from Data Centre in different Geographical location
1.	UB1 (p1)	50.41	250.98
2.	UB2 (p2)	53.04	299.55
3.	UB3 (p3)	58.29	310.95

Table 5.8: Response Time for user base served with same and different GL with some priority.

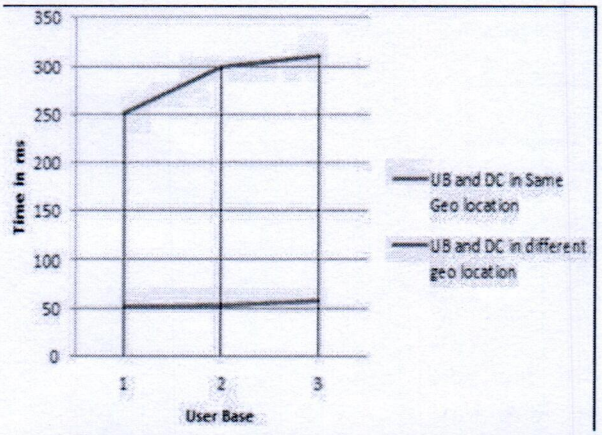


Figure 5.3: Graph showing improvement in response time or priority users

5.6 CONCLUSION

In today's situation it is obligatory for cloud service providers to have great relationship with their valued client. To maintain this relationship they need to serve their client needs in most efficient manners.

This paper briefs about the priority based load balancing approach that have a numerous advantages over traditional approach. The whole paper is divided into three basic sections. Section one depicts about the basic of load balancing and different load balancing approach. Section two focuses on proposed methodology and flowchart. Methodology depicts how priority is assigned to the users and datacenter and how these valued customers are served effectively and efficiently. Section three sheds lights on the proposed algorithm steps. When a customer sends a request for a cloud access then that customer can be provided services from that geographical area on the basis of his priority. Serving the customer on the basis of geographical locations in turn reduces the response time and hence increases the efficiency of the cloud.

REFERENCES

[1] Khiyaita, A., Zbakh, M., Bakkali, H. El., and Kettani, D.El, 2012, "Load balancing cloud computing: state of art," In National Days of Network Security and Systems (JNS2), IEEE, pages106–109

[2] Seungmin Kang, Bharadwaj Veeravalli, Khin Mi Mi Aung," Scheduling Multiple Divisible Loads in a Multi-cloud System", In 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing,USA, 2014, Page(s):371 – 378.

[3] Q. Cao, B. Wei and W. M. Gong, "An optimizd algorithm for task scheduling based on activity based costing in cloud computing," In International Conference on eSciences Modeling and Simulation Design. AK Peters Ltd ,2009, pp. 1-3.

[4] Chaczko Z, Mahadevan V, Aslanzadeh S, Mcdermid C. Availabil-ity and load balancing in cloud computing. In International Confer-ence on Computer and Software Modeling, Singapore 2011 Sep 16 (Vol. 14)

[5] Monika Choudhary, Sateesh Kumar Peddoju "A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment" IJERA ISSN: 2248-9622 Vol. 2, Issue 3, May-Jun 2012, pp.2564-2568.

[6] S. Singh and K. Kant, "Greedy grid scheduling algorithm in dynamic job submission environment," In International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT), 2011, pp. 933-936.

[7] L. Kuan, R. Yahyapour, P. Wieder, C. Kotsokalis, E. Yaqub, A. Jehangiri, "QoS-aware VM placement in multi-domain service level agreements scenarios", Proc. Int. Conf. on Cloud Computing, pp. 661-668, 2013

[8] Mohammadreza Mesbahi,Amir Masoud Rahmani,Anthony Theodore Chronopoulos "Cloud Light Weight : a New solution for Load Balancing in Cloud Computing",In IEEE International Conference on Data Science & Engineering (ICDSE),Kerla,pp44-50,2014.

[9] Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing",In 10 Proceedings of the 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops,USA,pp 551-556, 2010.

[10] Hung-Chang Hsiao, Hsueh-Yi Chung, Yu-Chang Chao, "Load Rebalancing for Distributed File Systems in Clouds", IEEE transactions on parallel and distributed systems, vol. 24, no. 5, pp 951-962, may 2013.

[11] Amandeep, Vandana Yadav, Faz Mohammad, "Different Strategies for Load Balancing in Cloud Computing Environment: a critical Study" , International Journal of Scientific Research Engineering & Technology (IJSRET), Volume 3, Issue 1,pp 52-56 April 2014.

[12] Zhen Xiao, Weijia Song, Qi Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment" , IEEE transactions on parallel and distributed systems, vol. 24, no. 6, pp 1107-1117, june 2013.

[13] Preeti Kushwah "A Survey on Load Balancing Techniques Using ACO Algorithm", International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5 (5),pp 6310-6314 , 2014.

[14] M. Thenmozhi, N. Tamilarasi "Oppurtunistic Routing Through Delay Analytical Methods in Ad-Hoc Wireless Networks"International journal of Computer Science and Engineering, Vol.6 , Issue.11 , pp.246-253, Nov-2018.

CHAPTER 6
LOAD BALANCING IN CLOUD USING HYBRID APPROACH

6.1 INTRODUCTION

The cloud computing is the innovation utilized over the Web. Cloud computing depends on the distributed computing giving versatile, hardware, software over the Internet. Cloud offers resources, for example, storage, network, and so on. It is the accumulation of servers appropriated over the all-regions of the world which give distinctive administrations on request. In the cloud registering, actual processing is finished by the remote servers. A cloud can be a public, private, or hybrid sort. These clouds give distinctive services to the end-clients on the request. The most basic ad-ministrations are Software as a Service (Saas), Platform as a Ser-vice (PaaS) and Infrastructure as a Service (IaaS) [1]. The private clouds are claimed by an association they have limited foundation and administrations they are secured from the outside world. While the general population clouds offer extensive scale foundation and administrations to the customer on the payment basis. Access to the public cloud is boundless it can be gotten to from any piece of the world; public clouds are highly inclined to security control. The third sort of cloud is Hybrid cloud they are framed by the blend of public and private clouds. Hybrid cloud gives adaptability in exchange offs between the cost-reserve funds and versatility of the public cloud and control of information assets at a private cloud. Numerous merchants offer cloud services, for example, Amazon, AT and T, Eucalyptus System, Google, Microsoft and some more [1][2]Cloud. The primary goal of cloud computing are to share hardware resources, software resources, data resources through web to decreased general cost of the framework, to give better execution as far as normal response time and normal processing time, keep up the framework consistency and to give future adjustment in the system [3].Basically there are diverse difficulties that should be dealt with like rapid demand

LB is the method of dissemination of workload among nodes in an appropriate way with the end goal that it guarantees no node in the framework is over-burden or sits idle. Cloud

LB diminishes costs and augments resource accessibility. An effective LB algorithm will ensure that each node in the framework accomplishes pretty much same volume of work. LB is an essential part since we can't anticipate the quantity of requests that are issued at each second in the cloud environment. The reason for LB in the cloud computing is in apportioning the load progressively among the nodes with a specific end goal to fulfil the client prerequisites and to give throughput by declaring the general accessible load to particular nodes [4][5][6]. Different LB algorithms are studied and then got the insight to combine few algorithms features to generate good algorithm for VM LB. In this paper we clubbed the features of two algorithms that are priority and ESCE [5][7].LB works on the principal of moving the load from lightly heavily loaded VM's of lightly loaded or no loaded VM's to make the system more stable and efficient.

6.2. EXISTING LOAD BALANCING ALGORITHMS

To distribute workload among different VM's efficiently in terms of throughput and latency we use:

6.2.1 Equally Spread Current Execution Algorithm (ESCE)

The LB balancer makes an attempt to safeguard equivalent load to any or all the VM's connected with the data centers (DC). It appropriates the heap to VM by checking the heap at current time and trade of the heap to that VM which is less stacked and result in less time taken, and gives high throughput. This Load balancer keeps up a record table of VM's and furthermore number of solicitations starting at now distributed to the VM [8]. Exactly when all the VM's are stacked and if the request originates from the DC to assign the new VM, it checks the run down table for least stacked VM. The DC gives the request to the VM perceived. Right when VM completes the errand, it is educated to DC which is furthermore told by the load balancer.

6.2.2. Priority Algorithm

In priority algorithms some priority is decided on which the tasks are executed. In our approach we have used double priority [9]. First priority is on cloudlets length that is the cloudlet with the smallest length will be given highest priority. Second, is on VM's that

are the VM's will be sorted on the basis of their MIPS. The cloudlet with the smallest length will be allocated to the VM with lowest MIPS. In this way the overall problem of starvation can also be avoided and response time can also be minimized as cloudlet with large length will be assigned VM with high priority. The prescribed calculation will endeavor to support the execution by giving the assets on ask for, which may achieve increase in number of assignment executions and subsequently decreasing the dismissal in the amount of employments submitted. One of the estimations is picked by stack balancer is illustrated in figure 5.1.

6.3 HYBRID ALGORITHM

The proposed algorithm is a combination of two algorithms i.e. Double priority [9, 13] and ESCE [12]. The algorithm constitutes the advantage of both. Firstly a decision tree is made indicating status of VM. Then priority algorithm is used when a series of client request are received. The re-quests are then broken into cloudlets and their length is calculated. Then the cloudlets are sorted on the basis of their length. Again VM's are also sorted on the basis of their MIPS and are allocated to cloudlets in first come first serve (FCFS) order such that cloudlet with smallest length receives VM with least MIPS. Using this technique the problem of starvation is avoided and total time taken to execute the cloudlet will also be minimized. Now we use the ESCE algorithm to distribute the load equally among all the servers so that no server is under loaded, overloaded or sitting idle. The whole scenario is demonstrated in figure 5.2.

Step 1: Initially set the status of all VM's to available and the decision tree has no entry.

Step 2: Again the data center controller will create VM's on the basis of their MIPS and will store them in the same way.

Step 3: The data center controller receives a new task.

Step 4: It will break the task into cloudlets and will calculate their length.

Step 5: Now the data center will sort the cloudlets on the basis of their length and will assign priority.

Step 6: Now allocate the sorted cloudlets to the list of VM's both in FCFS order.

Step 7: The load balancer will check if there is any VM which is under loaded, overloaded or sitting idle.

Step 8: If found, then load will be transferred from over-loaded and to under loaded or sitting idle.

Step 9: Update the decision table.

Step 10: Finish the execution.

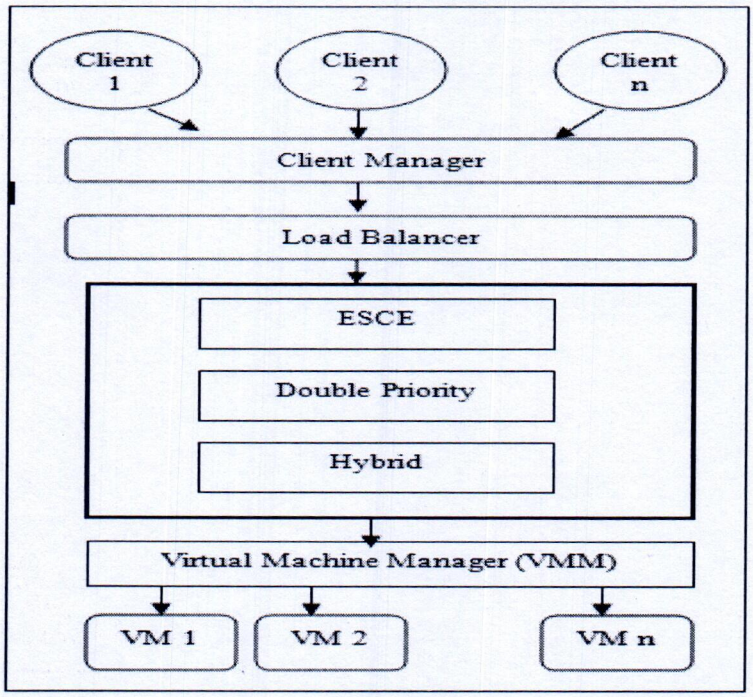


Figure 6.1: Load balancer with algorithms

6.4 SIMULATION ENVIRONMENT

Simulation remains for making a domain which looks and carries on like unique one. The proposed procedure or algorithm can be examined in a simulation environment. By utilizing the effectiveness, execution and so forth can be examined. It is more profitable to clients since they can watch their algorithm before executing it on real condition. Additionally, it decreases general cost as changes can be made before acknowledging it really.

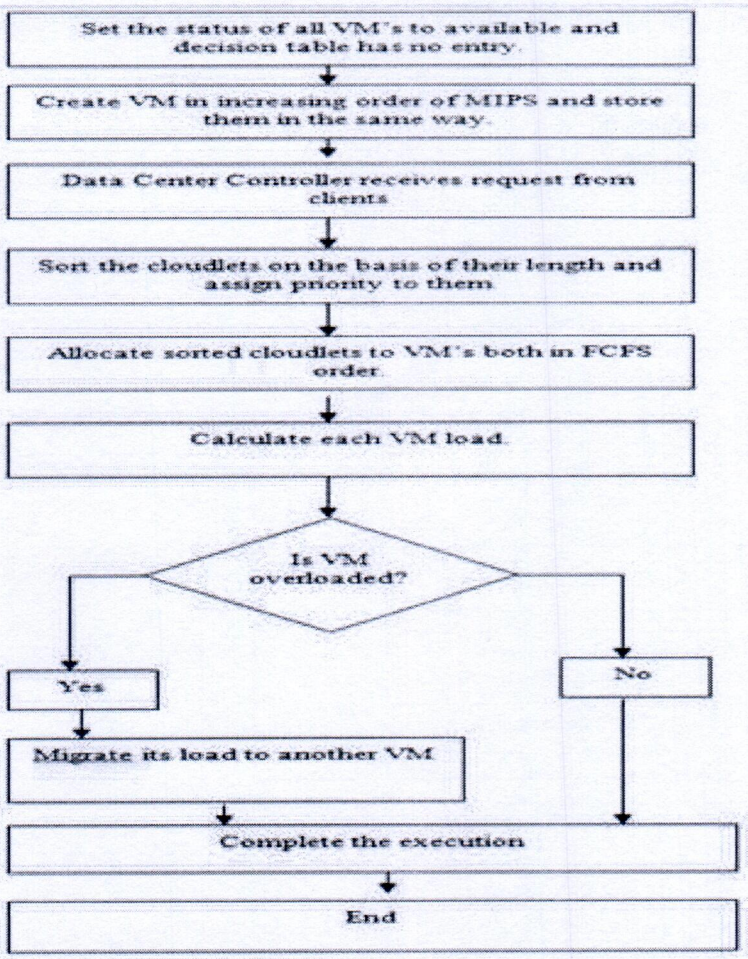

Set the status of all VM's to available and decision table has no entry.
Create VM in increasing order of MIPS and store them in the same way.
Data Center Controller receives request from clients
Sort the cloudlets on the basis of their length and assign priority to them
Allocate sorted cloudlets to VM's both in FCFS order.
Calculate each VM load.
Is VM overloaded?
Yes
No
Migrate its load to another VM
Complete the execution
End

Figure 6.2: Flowchart of hybrid algorithm

6.4.1 CloudSim

CloudSim objective is to give a reproduction situation that engages showing, recreation, and experimentation of creating Cloud application, empowering its customers to eye on distinct structure setup aftereffect that they have to inspect, without getting stressed over the lowered purposes of intrigue related to Cloud-based establishments and organizations. Remembering the true objective to look at changed load adjusting calculations outline of the distinctive sections ought to be set [10]. We can design our own dataset for different customer and different servers. By using CloudSim, researchers and industry-construct specialists can amass in light of specific structure design issues that they have to

investigate, without getting stressed over the low level purposes of premium related to Cloud-based systems and organizations

Distinctive functionalities of CloudSim are:

1. Support simulation of huge scale cloud computing data centers.
2. Support simulation and demonstrating of virtual server host.
3. Support simulation and demonstrating of computational resources.
4. Support simulation and demonstrating of datacenters.
5. Support for user defined scheduling policies.

6.4.2 Eclipse

Eclipse is a Integrated Development Environment (IDE) for growing principally with Java [2]. It contains workspace where programmer can do his programming. Mostly it is composed in Java and can run on Windows, Linux, and many other platforms having the support of JVM. In Eclipse Platform user can design an application using the concepts of modules. Applications developed using Eclipse Platform (counting the Eclipse IDE itself) can also be re-used by other developers [11].It also provide a good sup-port for additional plugins.

6.5 RESULTS AND DISCUSSION

The whole scenario is implemented in CloudSim 3.0 After implementing the proposed hybrid algorithm the following results are achieved.

From figure 5.3 it is clearly seen that the MIPS of all VM are produced randomly but in such a way that VM 0 will have the least MIPS while the last VM will have the maximum MIPS. Doing so the problem of starvation can be avoided. In figure 5.6, it is clearly demonstrated that no VM is under loaded, overloaded or sitting idle. All VM's are equally utilized to approximately 73%. Hence the proposed hybrid algorithm handles the problem of LB. In figure 5.5, it is clearly demonstrated that no VM is under loaded, overloaded or sitting idle. All VM's are equally utilized to approximately 73%. Hence the proposed hybrid algorithm handles the problem of load LB.

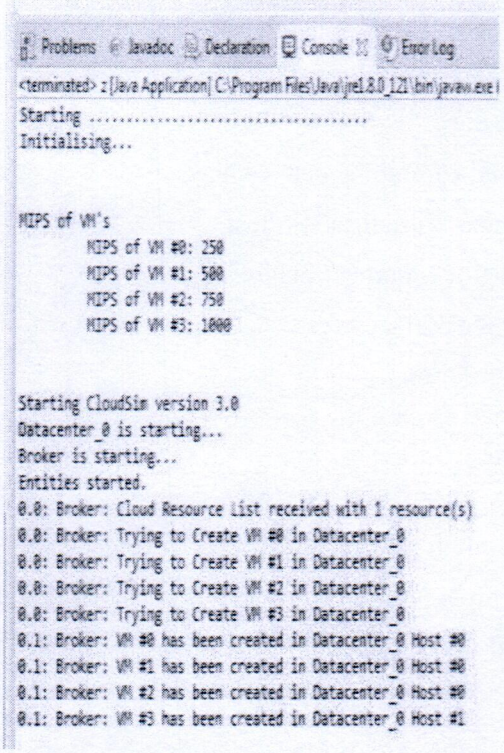

Problems Javadoc Declaration Console Error Log
<terminated> z [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe
Starting .....
Initialising...
MIPS of VM's
MIPS of VM #0: 250
MIPS of VM #1: 500
MIPS of VM #2: 750
MIPS of VM #3: 1000
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter_0 Host #0
0.1: Broker: VM #1 has been created in Datacenter_0 Host #0
0.1: Broker: VM #2 has been created in Datacenter_0 Host #0
0.1: Broker: VM #3 has been created in Datacenter_0 Host #1

Figure 6.3: VM's in increasing order of MIPS

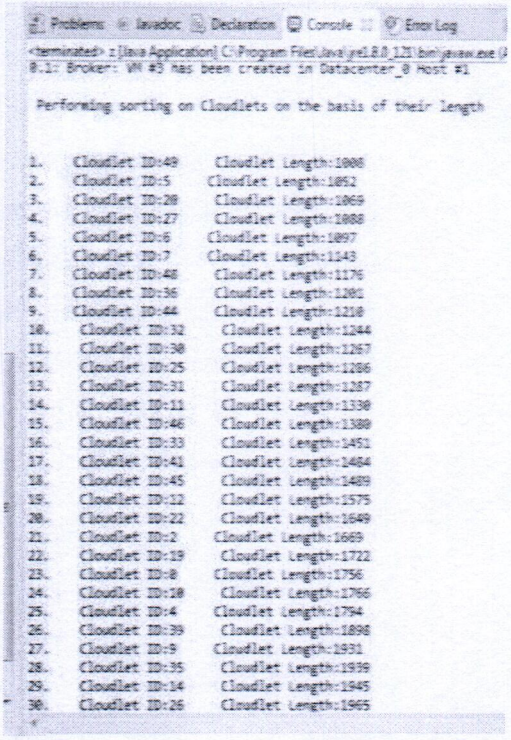

Problems Javadoc Declaration Console Error Log
0.1: Broker: VM #3 has been created in Datacenter_0 Host #1
Performing sorting on Cloudlets on the basis of their length
1. Cloudlet ID:49 Cloudlet Length:1000
2. Cloudlet ID:5 Cloudlet Length:1052
3. Cloudlet ID:20 Cloudlet Length:1069
4. Cloudlet ID:27 Cloudlet Length:1088
5. Cloudlet ID:6 Cloudlet Length:1097
6. Cloudlet ID:7 Cloudlet Length:1143
7. Cloudlet ID:48 Cloudlet Length:1176
8. Cloudlet ID:36 Cloudlet Length:1201
9. Cloudlet ID:44 Cloudlet Length:1218
10. Cloudlet ID:32 Cloudlet Length:1244
11. Cloudlet ID:30 Cloudlet Length:1267
12. Cloudlet ID:25 Cloudlet Length:1286
13. Cloudlet ID:31 Cloudlet Length:1287
14. Cloudlet ID:11 Cloudlet Length:1330
15. Cloudlet ID:46 Cloudlet Length:1380
16. Cloudlet ID:33 Cloudlet Length:1451
17. Cloudlet ID:41 Cloudlet Length:1464
18. Cloudlet ID:45 Cloudlet Length:1489
19. Cloudlet ID:12 Cloudlet Length:1575
20. Cloudlet ID:22 Cloudlet Length:1649
21. Cloudlet ID:2 Cloudlet Length:1669
22. Cloudlet ID:19 Cloudlet Length:1722
23. Cloudlet ID:0 Cloudlet Length:1756
24. Cloudlet ID:10 Cloudlet Length:1766
25. Cloudlet ID:4 Cloudlet Length:1794
26. Cloudlet ID:39 Cloudlet Length:1898
27. Cloudlet ID:9 Cloudlet Length:1931
28. Cloudlet ID:35 Cloudlet Length:1939
29. Cloudlet ID:14 Cloudlet Length:1945
30. Cloudlet ID:26 Cloudlet Length:1965

Figure 6.4: Performing sorting on cloudlets

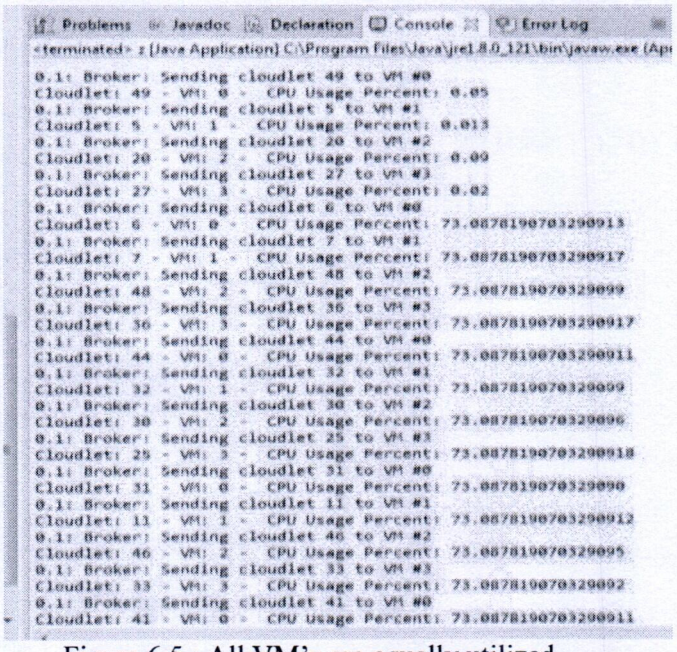
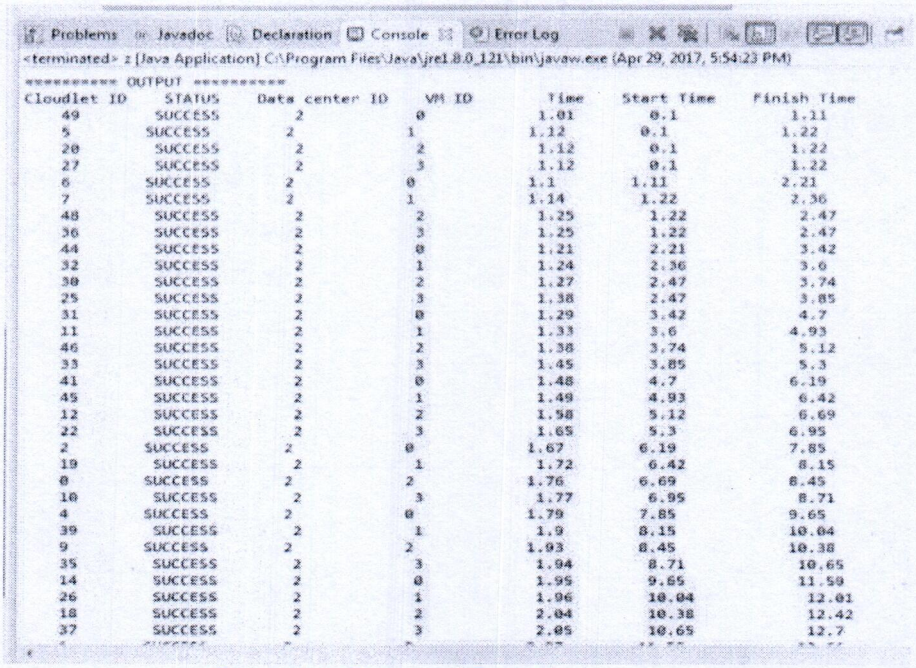

Problems Javadoc Declaration Console Error Log
<terminated> z [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe
0.1: Broker: Sending cloudlet 49 to VM #0
Cloudlet: 49 - VM: 0 - CPU Usage Percent: 0.05
0.1: Broker: Sending cloudlet 5 to VM #1
Cloudlet: 5 - VM: 1 - CPU Usage Percent: 0.013
0.1: Broker: Sending cloudlet 20 to VM #2
Cloudlet: 20 - VM: 2 - CPU Usage Percent: 0.09
0.1: Broker: Sending cloudlet 27 to VM #3
Cloudlet: 27 - VM: 3 - CPU Usage Percent: 0.02
0.1: Broker: Sending cloudlet 6 to VM #0
Cloudlet: 6 - VM: 0 - CPU Usage Percent: 73.08781907032909 13
0.1: Broker: Sending cloudlet 7 to VM #1
Cloudlet: 7 - VM: 1 - CPU Usage Percent: 73.08781907032909 17
0.1: Broker: Sending cloudlet 48 to VM #2
Cloudlet: 48 - VM: 2 - CPU Usage Percent: 73.08781907032909 9
0.1: Broker: Sending cloudlet 36 to VM #3
Cloudlet: 36 - VM: 3 - CPU Usage Percent: 73.08781907032909 17
0.1: Broker: Sending cloudlet 44 to VM #0
Cloudlet: 44 - VM: 0 - CPU Usage Percent: 73.08781907032909 11
0.1: Broker: Sending cloudlet 32 to VM #1
Cloudlet: 32 - VM: 1 - CPU Usage Percent: 73.08781907032909 9
0.1: Broker: Sending cloudlet 30 to VM #2
Cloudlet: 30 - VM: 2 - CPU Usage Percent: 73.08781907032909 6
0.1: Broker: Sending cloudlet 25 to VM #3
Cloudlet: 25 - VM: 3 - CPU Usage Percent: 73.08781907032909 18
0.1: Broker: Sending cloudlet 31 to VM #0
Cloudlet: 31 - VM: 0 - CPU Usage Percent: 73.08781907032909 0
0.1: Broker: Sending cloudlet 11 to VM #1
Cloudlet: 11 - VM: 1 - CPU Usage Percent: 73.08781907032909 12
0.1: Broker: Sending cloudlet 46 to VM #2
Cloudlet: 46 - VM: 2 - CPU Usage Percent: 73.08781907032909 5
0.1: Broker: Sending cloudlet 33 to VM #3
Cloudlet: 33 - VM: 3 - CPU Usage Percent: 73.08781907032909 2
0.1: Broker: Sending cloudlet 41 to VM #0
Cloudlet: 41 - VM: 0 - CPU Usage Percent: 73.08781907032909 11

Figure 6.5 : All VM's are equally utilized



Problems | Javadoc | Declaration | Console | Error Log

<terminated> z [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Apr 29, 2017, 5:54:23 PM)

********** OUTPUT **********

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
49	SUCCESS	2	0	1.01	0.1	1.11
5	SUCCESS	2	1	1.12	0.1	1.22
20	SUCCESS	2	2	1.12	0.1	1.22
27	SUCCESS	2	3	1.12	0.1	1.22
6	SUCCESS	2	0	1.1	1.11	2.21
7	SUCCESS	2	1	1.14	1.22	2.36
48	SUCCESS	2	2	1.25	1.22	2.47
36	SUCCESS	2	3	1.25	1.22	2.47
44	SUCCESS	2	0	1.21	2.21	3.42
32	SUCCESS	2	1	1.24	2.36	3.6
30	SUCCESS	2	2	1.27	2.47	3.74
25	SUCCESS	2	3	1.38	2.47	3.85
31	SUCCESS	2	0	1.29	3.42	4.7
11	SUCCESS	2	1	1.33	3.6	4.93
46	SUCCESS	2	2	1.38	3.74	5.12
33	SUCCESS	2	3	1.45	3.85	5.3
41	SUCCESS	2	0	1.48	4.7	6.19
45	SUCCESS	2	1	1.49	4.93	6.42
12	SUCCESS	2	2	1.58	5.12	6.69
22	SUCCESS	2	3	1.65	5.3	6.95
2	SUCCESS	2	0	1.67	6.19	7.85
19	SUCCESS	2	1	1.72	6.42	8.15
0	SUCCESS	2	2	1.76	6.69	8.45
10	SUCCESS	2	3	1.77	6.95	8.71
4	SUCCESS	2	0	1.79	7.85	9.65
39	SUCCESS	2	1	1.9	8.15	10.04
9	SUCCESS	2	2	1.93	8.45	10.38
35	SUCCESS	2	3	1.94	8.71	10.65
14	SUCCESS	2	0	1.95	9.65	11.59
26	SUCCESS	2	1	1.96	10.04	12.01
18	SUCCESS	2	2	2.04	10.38	12.42
37	SUCCESS	2	3	2.05	10.65	12.7

Figure 6.6: Output of Execution

In figure 5.6, the output of the proposed algorithm is demonstrated and it clearly seen that the total time taken to complete the task is reduced to a large amount. Now results of hybrid algorithms are compared with other two algorithms with same set of inputs shown in table 6.1.Those results are plotted in figure 6.7, and it is concluded that overall response time of hybrid algorithms is improved about 30 to 40 percent as compared with other algorithms like ESCE and priority.

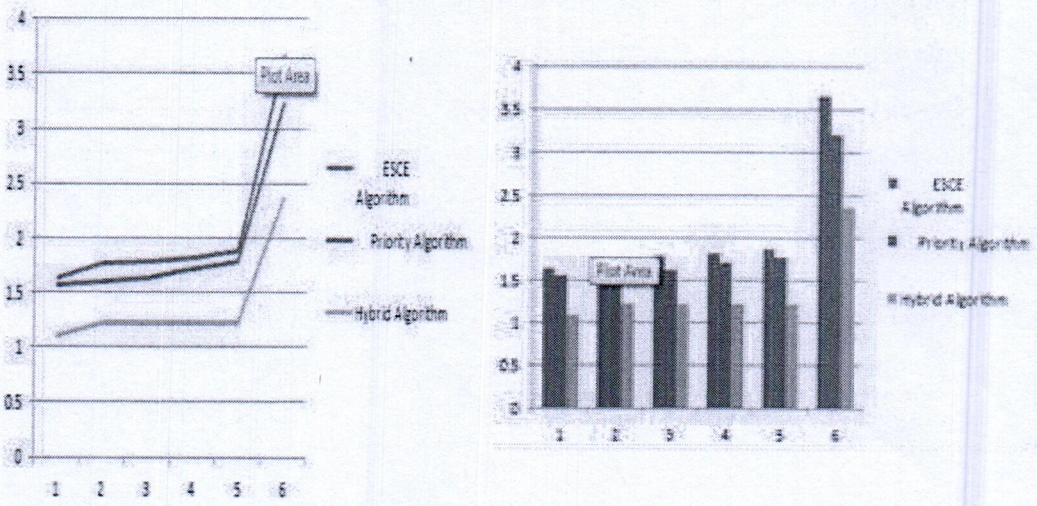


Figure 6.7: Results comparison using line graph and bar graph

Cloudlet ID in sorted order of their length	Cloudlet size (kilobytes)	Overall Response time(seconds)		
		ESCE Algorithm	Priority Algorithm	Hybrid Algorithm
49	1008	1.64	1.56	1.10
5	1052	1.76	1.60	1.22
20	1069	1.79	1.63	1.22
27	1088	1.82	1.71	1.22
6	1097	1.88	1.78	1.22
7	1143	3.66	3.22	2.36

Table 6.1 Result comparison with other algorithms

6.6 CONCLUSION

Hybrid approach is proposed using double priority and equally spread current execution algorithms and then implemented in CloudSim 3.0 environment. The algorithm proposed takes advantages of both the above mentioned algorithms. Here overall response time and finish time are considered as evaluation parameters. From the results it can be clearly observed that overall response time and processing time is less for hybrid approach as compared to others. In future some more parameters can also be taken for evaluation and priority can be assigned on some other factors of cloudlets such as total cost to execute a cloudlet. In our approach we have considered MIPS of VM's as an important factor but in future some other factors like bandwidth, memory can also be taken into account.

REFERENCES

[1] Ranjan Dinesh, Canino Anthony, Izaguirre A Jesus and Douglas Thain "Converting a High Performance Application to an Elastic Cloud Application" 3 rd IEEE International Conference on Cloud Computing Technology and Science, Nov 11.

[2] A. Y. Zomaya,& Y. H. Teh. (2014). Observations on using genetic algorithms for dynamic load-balancing. IEEE Transaction on Parallel and Distributed Systems, vol. 12, no. 9, pp. 899-911.

[3] Buyya, Rajkumar.,Broberg, James., Goscinski, Andrzej. "Cloud Computing Principles and Paradigms" (1sted.). Hoboken, New Jersey, USA: Wiley, 2011.

[4] R. N. Calheiros and R. Buyya, "Meeting deadlines of scientific workflows in public clouds with tasks replication," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 7, pp. 1787–1796, 2014.

[5] Eddy Caron , Luis Rodero-Merino —Auto-Scaling , Load Balancing And Monitoring In Commercial And OpenSource Clouds — Research Report, January2012.

[6] Mishra ,Ratan , Jaiswal, Anant,P—Ant Colony Optimization: A Solution Of Load Balancing In Cloud‖,April 2012, International Journal Of Web & Semantic Technology;Apr2012, Vol. 3 Issue 2, P33.

[7] R. Basker, V. RhymendUthariaraj, and D. Chitra Devi, "An enhanced scheduling in weighted round robin for the cloud infrastructure services," International Journal of Recent Advance in Engineering & Technology, vol. 2, no. 3, pp. 81–86, 2014.

[8] BhathiyaWickremasinghe ,Roderigo N. Calherios Cloud Analyst: A Cloud-Sim-Based Visual Modeler For Analyzing Cloud Computing Environments And Applications‖. Proc Of IEEE International Conference On Advance Information Networking And Applications ,2010.

[9] GenaudStephane and GossaJulien "Cost-wait Tradeoffs in Client-side Resource Provisioning with Elastic Clouds", IEEE 4th International Conference on Cloud Computing, 2011.

[10]R.N. Calheiros, R. Ranjan, A. Beloglazov, C. Rose, R. Buyya, "Cloudsim:A for modeling and simulation of cloud computing environ- ments and evaluation of resource provisioning algorithms",inSoftware:PracticeandExperience(SPE),Vol:41,No:1,ISSN:00380644,Wiley Press,USA,pp:23-50,2011.

[11] Buyya R, Ranjan R, Calheiros R N. "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities" Proceedings of the Conference on High Performance Computing and Simulation (HPCS 2009), Leipzig, Germany. IEEE Press: New York, U.S.A., 21–24 June 2009; 1–11.

[12] Dr.S. Suguna and R. Barani," Simulation of Dynamic Load Balancing Algorithms", Bonfring International Journal of Software Engineering and Soft Computing, Vol. 5, No.1, July 2015.

[13] Ishwari Singh Rajput, Deepa Gupta, "A Priority Based Round Robin CPU Scheduling Algorithm for Real Time Systems", *IJIET*, vol. 1, no. 3, October 2012, ISSN 2319-1058.

CHAPTER 7
CONCLUSION AND FUTURE WORK

This thesis gives the introduction of Cloud computing and background of various load balancing techniques to balance the workload on to the cloud server.

Started with the highlights of virtualization technology, its application in cloud and advantages reaped in cloud computing. Firstly the theses gives the introduction about, why cloud computing is required and is essential for IT organizations. Here it focuses on the matrices like throughput, response time etc. which need to be considered to improve the efficiency of the cloud.

Further focus will be on different load balancing techniques used in the cloud. Basically load balancing techniques will be divided into two types as static and dynamic. Static load balancing techniques are good for systems where there is initial information basic load and the configurations of the systems. Some of them discussed here are round robin, min-min, max-min etc. All these algorithms have some limitations related to response time and not able to tackle real time load variations.

Furthermore dynamic load balancing techniques are discussed. Some of them are Ant colony optimization and honey bee forging techniques. These techniques also have issues related to throughput, response time etc. These dynamic load balancing techniques overcome some of the issues related to static techniques.

One of the chapters has discussion on comparison of load balancing and re-balancing techniques. Load re-balancing approach is also one of the good solutions to balance the load with minimal rebalancing time or load migration time as compared to load balancing technique.

Performances of algorithms are analyzed qualitatively and following conclusion can be made from the work:

1. Load re-balancing is also one of the good ways to balance the load on the cloud servers efficiently.
2. Proposed hybrid load balancing technique shows better results as compared to priority and ESCE algorithms.

FUTURE WORK

Research and development are continuous process. Every completed research work gives birth to new challenges for further improvements. As per the work done so far in this research only few performance metrics like response time and availability of resources are covered.

Lot more future scope is there in the direction of better resource utilization that come comes under green cloud technology.

MAPPING OBJECTIVES WITH PUBLICATIONS

S. NO	OBJECTIVES	PUBLICATIONS
1	To proposed the load rebalancing approach for minimizing the load movement time and response time.	Manmohan Sharma, Anil Kumar, Sourabh Singh Verma" Load re-balancing approach for distributed load in cloud environment "International Journal of latest trends in Engineering and Technology"(IJLTET), Special Issue IC3NS 2018, pp. 019-022.
2	To prioritize cloud customer based on services demand and geographical location for better customer satisfaction	Manmohan Sharma, Vinod Kumar Jain "Load balancing in cloud using prioritization based on Quality of Services (QoS) demand", International Journal of Computer Sciences and Engineering (IJCSE),Vol 6, Issue 11, Nov 2018, pp 938-943.
3	To improve the resource utilization in a cloud system using better load balancing techniques.	• Manmohan Sharma, VK Jain "Load Balancing in Cloud using Hybrid Approach", International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol.8, Issue 6C,April 2019, pp 9-13. • Manmohan Sharma, Anil Kumar "Load balancing in Cloud using modified genetic algorithm", International Journal of Computer Science and Information Security (IJCSIS), Vol. 15, No. 9, September 2017